SYNWAY®

三汇系列语音卡

程序员手册

Version 5.0.0.0

杭州三汇信息工程有限公司

www.sanhuid.com

目 录

目 录 ..... **i**
版权声明 ..... **xxviii**
SynCTI驱动程序授权协议 ..... **xxix**
版本修订记录 ..... **xxx**
前 言 ..... **xxxi**
1 Synway板卡编程基础 **1**
1.1 Synway板卡的分类（CTI系列） 1
1.2 Synway板卡的分类（REC系列） 2
1.3 SynCTI支持的CODEC 3
1.3.1 编码格式转换的工具函数 *3*
1.3.2 板卡支持的CODEC *4*
1.4 SynCTI驱动程序支持的操作系统 5
1.5 SynCTI 驱动程序的体系结构（CTI系列） 5
1.6 SynCTI 驱动程序的体系结构（REC系列） 7
1.7 SynCTI 驱动程序的配套工具软件 8
1.7.1 系统配置工具软件ShCtiConfig.exe *8*
1.7.2 信号音分析工具软件ShTA.exe *9*
1.7.3 SS7 信令服务器配置程序SS7Cfg.exe *9*
1.7.4 SS7 信令服务器SS7monitor.exe *9*
1.7.5 MSU解码工具软件MsuDecode.exe *9*
1.8 基本概念 9
1.8.1 通道（CTI系列） *9*
1.8.2 通道（REC系列） *10*
1.8.3 通道的逻辑编号 *10*
1.8.4 数字中继线 *10*
1.8.4.1 数字中继线的帧同步 11
1.8.4.2 数字中继线的时钟 11
1.8.4.3 数字中继线的配置 11
1.8.4.4 数字中继线中的信令时隙 11
1.8.4.5 数字中继线的编号 12
1.8.4.6 设置B信道的语音编码格式 12
1.8.5 板卡的用户授权号 *12*
1.8.6 模拟电话线的电压变化 *12*
1.8.7 模拟电话线上的铃流信号 *14*
1.8.8 模拟电话线上的闪断信号 *15*
1.8.9 模拟电话线上的主叫号码 *15*
1.9 配置应用系统的时钟 16
1.10 语音处理 17
1.10.1 语音播放（CTI系列） *17*

1.10.1.1 设置放音数据音量及去向 17
1.10.1.1.1 CTI系列 17
1.10.1.1.2 REC系列 17
1.10.1.2 设置播放任务的终止条件 18
1.10.1.3 播放预加载文件 18
1.10.1.4 播放单个文件 19
1.10.1.5 播放文件序列 19
1.10.1.6 播放单个内存 20
1.10.1.7 播放内存序列 20
1.10.1.8 播放乒乓内存 20
1.10.1.9 终止播放任务的通用函数 21
1.10.1.10 获取播放任务的相关信息 21
1.10.2 录音操作 21
1.10.2.1 设置录音信号源 21
1.10.2.2 设置录音任务的终止条件 21
1.10.2.3 设置预录音功能 21
1.10.2.4 设置录音文件截尾功能 22
1.10.2.5 设置录音AGC 22
1.10.2.6 获取录音操作的信息 22
1.10.2.7 录音函数概述 23
1.10.2.7.1 文件录音 23
1.10.2.7.2 内存录音 23
1.10.2.7.3 乒乓缓冲区录音 23
1.10.3 DTMF检测器(DTMF Detector) 24
1.10.3.1 DTMF滤波器（DTMF Filter） 25
1.10.3.2 DTMF脉宽滤波器 25
1.10.3.3 WaitDtmf任务 26
1.10.3.4 Callback方式编程 26
1.10.3.5 检测脉冲方式的按键信号 26
1.10.4 DTMF发生器(DTMF Generator)（CTI系列） 26
1.10.5 信号音检测器(Tone Detector) 27
1.10.5.1 FFT模块 29
1.10.5.2 噪声滤波器(Noise Filter) 29
1.10.5.3 呼叫进程音检测器(Call Progress Tone Detector) (SHT系列only) 30
1.10.5.3.1 拨号音检测器（Dial Tone Detector） 31
1.10.5.3.2 回铃音检测器 31
1.10.5.3.2.1 忙音检测器 32
1.10.5.3.2.2 常规功能描述 32
1.10.5.3.2.3 背靠背忙音检测功能 33
1.10.5.3.3 自定义信号音检测器（User-defined Tone Detector） 34
1.10.5.4 传真进程音检测器(CTI系列) 34
1.10.5.4.1 简易忙音检测器(Simplified Busy Tone Detector) 35
1.10.5.5 SIT检测器(Special Information Tone Detector) 36
1.10.5.6 AMD (Answering Machine Detector) 36

1.10.6 信号音发生器(Tone Generator)（CTI系列） *36*
1.10.6.1 设置信号音发生器的工作参数 36
1.10.6.2 产生信号音 37
1.10.7 回波抵消器（CTI系列） *37*
1.10.8 Barge in检测器（Barge in Detector） *37*
1.10.9 TDM交换总线 *38*
1.10.10 分布式电话会议(CTI系列) *39*
1.10.10.1 基本概念 39
1.10.10.2 管理会议室 41
1.10.10.3 管理会议通道 42
1.10.10.4 给会议室播放背景音乐 42
1.10.10.5 录制会议室的声音 42
1.11 FSK数据收发器（CTI系列） 43
1.11.1 FSK数据发送器 *43*
1.11.2 FSK数据接收器 *43*
1.12 传真（CTI系列） 44
1.12.1 设置传真资源通道的数量 *44*
1.12.2 传真操作支持的速率 *44*
1.12.3 传真操作支持的文件格式 *45*
1.12.4 设置传真操作的工作参数 *46*
1.12.5 获取传真操作的信息 *46*
1.12.6 发送传真 *46*
1.12.7 接收传真 *47*
1.13 板载音频功率放大器 47
1.14 编写基于Synway板卡的应用程序 47
1.14.1 设置驱动程序的事件输出模式 *47*
1.14.1.1 事件等待方式的编程示例 48
1.14.1.2 事件回调方式的编程示例 49
1.14.2 输出事件的数据结构 *51*
1.14.2.1 MESSAGE_INFO 51
1.14.2.2 SSM_EVENT 55
1.14.3 事件的过滤 *57*
1.14.4 使用自定义事件 *57*
1.14.5 常见编程平台下的编程要点 *57*
1.14.5.1 MS VC/C++ 57
1.14.5.2 VB 57
1.14.5.3 C++ BUILDER 58
1.14.5.4 Delphi 59
1.14.5.5 PB6.5 59
1.14.5.6 其它 59
1.14.5.7 Linux操作系统 59
1.14.6 配套演示程序（CTI系列） *60*
1.14.7 配套演示程序（REC系列） *60*
1.14.8 使用驱动程序提供的定时器 *60*

1.14.9 使用驱动程序提供的调试功能 61
1.15 SHT系列板卡（CTI系列） 61
1.15.1 SHT系列板卡简介 61
1.15.2 SHT系列板卡的工作原理框图 62
1.15.3 模拟中继线通道 64
1.15.3.1 在模拟线路上产生闪断信号 64
1.15.3.2 检测极性反转信号 64
1.15.3.3 模拟中继线通道的状态转移 65
1.15.3.4 远端摘机检测器(Remote Pickup Detector)（SHT系列only） 67
1.15.3.4.1 普通的远端摘机检测器 68
1.15.3.4.2 增强的远端摘机检测器 70
1.15.4 坐席通道 70
1.15.4.1 状态迁移图 70
1.15.4.2 检测话机的摘机/挂机动作 71
1.15.4.3 检测话机的闪断信号 71
1.15.4.4 向电话机发送振铃信号 71
1.15.4.5 在电话线上产生极性反转信号 72
1.15.4.6 检测到话机摘机后自动送拨号音 72
1.15.5 联合模块 72
1.15.6 磁石电话机通道 72
1.15.6.1 概述 72
1.15.6.2 状态迁移图 73
1.15.7 使用未安装业务模块的通道 74
1.16 SHD系列板卡（CTI系列） 74
1.16.1 SHD系列板卡简介 74
1.16.2 SHD系列板卡的工作原理框图 75
1.16.3 设置来话呼叫的收号规则 78
1.16.4 SS7 协议的编程 78
1.16.4.1 OPC和DPC 78
1.16.4.2 时隙分配 78
1.16.4.3 设置地址信号中备用值的对应字符 79
1.16.4.4 SS7 信令应用系统的配置实例 80
1.16.4.5 TUP协议的编程与应用 81
1.16.4.5.1 TUP通道的状态转移 81
1.16.4.5.2 通道的闭塞与解除 86
1.16.4.5.2.1 通道闭塞的概念 86
1.16.4.5.2.2 通道闭塞的作用 86
1.16.4.5.2.3 通道闭塞的途径 86
1.16.4.5.2.4 实现通道闭塞的函数列表 87
1.16.4.5.3 电路的复原 87
1.16.4.5.4 “锁定主叫用户”功能 87
1.16.4.5.5 TUP协议的配置 87
1.16.4.6 ISUP协议的编程与应用 88
1.16.4.6.1 ISUP协议的配置 88

1.16.4.6.2 ISUP通道的状态转移 90
1.16.4.6.3 ISUP通道工作于汇接局方式 95
1.16.4.6.4 不使用SynCTI提供的ISUP协议状态机 96
1.16.4.7 SS7 的高级编程 96
1.16.4.7.1 使用MTP3 层提供的服务 96
1.16.4.7.2 使用SS7 服务器的客户端软件编程接口 96
1.16.4.7.3 使用SS7 服务器的虚电路编程接口 96
1.16.4.7.4 使用SCCP协议的编程接口 97
1.16.5 ISDN通道的编程 97
1.16.5.1 基本概念 97
1.16.5.1.1 网络侧模式与用户侧模式 97
1.16.5.1.2 TEI值 97
1.16.5.1.3 通路识别表示方法 97
1.16.5.1.4 端局方式与汇接局方式 97
1.16.5.1.5 ISDN号码与ISDN地址 98
1.16.5.2 ISDN通道的状态转移图 99
1.16.5.3 ISDN协议的配置 102
1.16.5.4 ISDN高级编程接口 104
1.16.6 SS1 通道编程指南 104
1.16.6.1 选择信令协议 104
1.16.6.2 设置接续过程的基本工作参数 104
1.16.6.3 设置状态机中通道的工作方式 104
1.16.6.4 中国 1 号信令的状态机描述 104
1.16.6.4.1 来话呼叫时启用号码拦截功能 104
1.16.6.4.2 与Dialogic公司SS1 通道对接 105
1.16.6.4.3 中国 1 号信令的状态转移 105
1.16.6.4.4 输出呼叫过程中的调试信息 109
1.16.6.5 Line side协议的状态机描述 110
1.16.6.6 SS1 通道的高级编程接口 111
1.16.6.7 No.1 信令的配置 111
1.16.7 连接到远端坐席接入模块 112
1.17 SHV系列板卡（CTI系列） 113
1.17.1 SHV系列板卡简介 113
1.17.2 SHV系列板卡的典型应用原理框图 113
1.17.3 有关变声功能的函数一览表 113
1.18 SHN系列板卡（CTI系列） 113
1.18.1 SHN系列板卡简介 113
1.18.2 SHN系列板卡的工作原理框图 114
1.18.3 SIP通道的编程 116
1.18.3.1 SIP编程基本概念 116
1.18.3.2 SIP通道的状态转移图 117
1.18.3.3 SIP通道的配置 120
1.18.4 H.323 通道的编程 120
1.18.4.1 H.323 编程基本概念 120

1.18.4.1.1 H.323 实体 120
1.18.4.1.2 H.323 消息 121
1.18.4.1.3 实体交互 121
1.18.4.1.3.1 终端直接呼叫，没有网守 122
1.18.4.1.3.2 两个终端在同一个网守上注册 122
1.18.4.2 H.323 通道的状态转移图 123
1.18.4.3 H.323 通道的配置 124
1.19 DST系列板卡（REC系列） 124
1.19.1 DST系列板卡简介 124
1.19.2 DST系列板卡支持的交换机型号 125
1.19.3 DST系列板卡的工作原理框图 132
1.19.4 DST系列板卡的配置 134
1.19.5 DST系列板卡的语音操作 134
1.19.6 DST系列板卡的信令编程 135
1.19.6.1 使用D信道的原始信令消息 135
1.19.6.2 使用驱动程序提供的通用呼叫状态机 136
1.19.6.2.1 DST通道的状态迁移图 136
1.19.6.2.2 如何获取有关电话号码的更多信息 137
1.19.6.2.3 如何获LCD上的显示信息 140
1.20 ATP系列板卡（REC系列） 140
1.20.1 ATP系列板卡简介 140
1.20.2 ATP系列板卡的工作原理框图 140
1.20.3 模拟中继线录音通道的状态迁移图 143
1.20.4 检测摘机/挂机状态 143
1.21 DTP系列板卡（REC系列） 144
1.21.1 DTP系列板卡简介 144
1.21.2 DTP系列板卡的工作原理框图 144
1.21.3 术语介绍 146
1.21.4 DTP系列板卡的语音功能 147
1.21.4.1 获取DTMF 147
1.21.4.2 语音录制 147
1.21.5 驱动程序实现的状态机 147
1.21.5.1 数字中继线录音通道的通用状态机模型 147
1.21.5.2 获取呼叫信息 148
1.21.6 DTP系列板卡的应用及配置实例 148
1.21.6.1 使用No.1 信令/ISDN PRI协议的系统 148
1.21.6.2 使用SS7 信令的系统 149
2 SynCTI开发包函数说明 151
2.1 系统函数 151
2.1.1 驱动平台初始化函数 151
2.1.1.1 SsmStartCti 151
2.1.1.2 SsmCloseCti 152
2.1.2 设置API和事件输出方式的函数 152
2.1.2.1 SsmSetLogOutput 152

2.1.3 有关定时器的函数 153
2.1.3.1 SsmStartTimer 153
2.1.3.2 SsmStopTimer 153
2.1.3.3 StartTimer 154
2.1.3.4 ElapseTime 154
2.1.4 获取系统错误信息的函数 155
2.1.4.1 SsmGetLastErrCode 155
2.1.4.2 SsmGetLastErrMsg 156
2.1.4.3 SsmGetLastErrMsgA 156
2.1.4.4 fBmp_GetErrMsg 156
2.1.5 获取板卡信息的函数 157
2.1.5.1 获取安装板卡的信息 157
2.1.5.1.1 SsmGetMaxCfgBoard 157
2.1.5.1.2 SsmGetMaxUsableBoard 157
2.1.5.1.3 GetTotalPciBoard 158
2.1.5.2 获取板卡授权号 158
2.1.5.2.1 SsmGetAccreditId 158
2.1.5.2.2 SsmGetAccreditIdEx 158
2.1.5.3 获取板卡型号 159
2.1.5.3.1 SsmGetBoardModel 159
2.1.5.3.2 GetPciBoardModel 160
2.1.5.4 获取板卡序列号 162
2.1.5.4.1 SsmGetPciSerialNo 162
2.1.5.4.2 GetPciBoardSerailNo 163
2.1.5.5 获取文件的版本信息 163
2.1.5.5.1 SsmGetDllVersion 163
2.1.6 获取通道属性的函数 164
2.1.6.1 SsmGetMaxCh 164
2.1.6.2 SsmGetChType 164
2.1.6.3 SsmGetChHdInfo 165
2.1.6.4 SsmGetAppChId 165
2.1.7 设置通道属性的函数 166
2.1.7.1 SsmSetFlag 166
2.1.7.2 SsmGetFlag 167
2.2 有关事件方式编程的函数 168
2.2.1 获取事件的函数 168
2.2.1.1 SsmWaitForEvent 168
2.2.1.2 SsmWaitForEventA 168
2.2.1.3 SsmGetEvent 168
2.2.1.4 SsmGetEventA 168
2.2.1.5 SsmGetInterEventType 169
2.2.2 过滤输出事件 170
2.2.2.1 SsmSetEvent 170
2.2.3 获取编程模式 173

2.2.3.1 SsmGetEventMode 173
2.2.4 通过驱动程序输出应用程序的自定义事件 173
2.2.4.1 SsmPutUserEvent 173
2.2.4.2 SsmPutUserEventA 173
2.3 有关呼叫状态机的函数 174
2.3.1 有关去话呼叫的函数（CTI系列） 174
2.3.1.1 搜索空闲通道 174
2.3.1.1.1 SsmSearchIdleCallOutCh 174
2.3.1.2 发起去话呼叫 175
2.3.1.2.1 SsmAutoDial 175
2.3.1.2.2 SsmAutoDialEx 175
2.3.1.2.3 SsmAppendPhoNum 178
2.3.1.2.4 SsmChkAutoDial 179
2.3.1.2.5 SsmGetAutoDialFailureReason 180
2.3.1.2.6 SsmSetTxCallerId 181
2.3.1.2.7 SsmGetTxCallerId 182
2.3.1.2.8 SsmSetTxOriginalCallerID 182
2.3.1.2.9 SsmSetWaitAutoDialAnswerTime 183
2.3.1.2.10 SsmGetWaitAutoDialAnswerTime 184
2.3.1.2.11 SsmGetKB 184
2.3.1.3 SS1 通道的专用函数 185
2.3.1.3.1 SsmSetKA 185
2.3.1.3.2 SsmSetKD 186
2.3.2 有关来话呼叫的函数 187
2.3.2.1 获取主叫用户信息 187
2.3.2.1.1 SsmChkOpCallerId 187
2.3.2.1.2 SsmGetCallerId 187
2.3.2.1.3 SsmGetCallerIdA 187
2.3.2.1.4 SsmGetCallerIdEx 187
2.3.2.1.5 SsmGetCallerName 188
2.3.2.1.6 SsmClearCallerId 189
2.3.2.1.7 SsmClearCallerIdEx 189
2.3.2.1.8 SsmGetKA 189
2.3.2.1.9 SsmGetKD 190
2.3.2.2 获取被叫用户信息 191
2.3.2.2.1 SsmGetPhoNumStr 191
2.3.2.2.2 SsmGetPhoNumStrA 191
2.3.2.2.3 SsmGetPhoNumLen 191
2.3.2.3 获取第 1 被叫号码信息 192
2.3.2.3.1 SsmGet1stPhoNumLen 192
2.3.2.3.2 SsmGet1stPhoNumStr 192
2.3.2.3.3 SsmGet1stPhoNumStrA 192
2.3.2.4 设置来话接续过程参数 193
2.3.2.4.1 SsmEnableAutoSendKB 193

2.3.2.4.2 SsmGetAutoSendKBFlag 194
2.3.2.4.3 SsmSetKB 194
2.3.3 通道状态迁移事件 196
2.3.3.1 SsmGetChState 196
2.3.3.2 SsmGetChStateKeepTime 198
2.3.4 获取PendingReason 198
2.3.4.1 SsmGetPendingReason 198
2.3.5 通道摘机函数（CTI系列） 200
2.3.5.1 SsmPickup 200
2.3.5.2 SsmPickupANX 200
2.3.6 通道挂机函数（CTI系列） 201
2.3.6.1 SsmHangup 201
2.3.6.2 SsmHangupEx 201
2.3.7 设置通道的呼叫方向 203
2.3.7.1 SsmSetAutoCallDirection 203
2.3.7.2 SsmGetAutoCallDirection 203
2.3.8 获取释放原因 204
2.3.8.1 SsmGetReleaseReason 204
2.3.9 TUP通道专用函数 205
2.3.9.1 SsmSetCalleeHoldFlag 205
2.3.9.2 SsmGetTupFlag 205
2.3.9.3 SsmSetTupParameter 206
2.3.10 ISUP通道专用函数 207
2.3.10.1 SsmSetISUPCAT 207
2.3.10.2 SsmSetIsupParameter 207
2.3.10.3 SsmGetIsupParameter 208
2.3.10.4 SsmSetIsupFlag 209
2.3.10.5 SsmGetIsupFlag 210
2.3.10.6 SsmGetIsupUPPara 211
2.3.10.7 SsmSetIsupUPPara 211
2.3.10.8 SsmSendIsupMsg 212
2.3.10.9 SsmGetRedirectionInfReason 213
2.3.10.10 SsmIsHaveCpg 213
2.3.10.11 SsmGetCpg 213
2.3.10.12 SsmIsupGetUsr 214
2.3.10.13 SsmIsupSendUsr 215
2.3.11 ISDN通道专用函数 215
2.3.11.1 设置Setup消息的参数 215
2.3.11.1.1 SsmISDNSetDialSubAddr 215
2.3.11.1.2 SsmISDNSetTxSubAddr 216
2.3.11.2 获取来话呼叫信息 216
2.3.11.2.1 获取子地址信息 216
2.3.11.2.1.1 SsmISDNGetSubAddr 216
2.3.11.2.1.2 SsmISDNGetCallerSubAddr 217

2.3.11.2.2 获取用户自定义主叫号码 217
2.3.11.2.2.1 SsmGetUserCallerId 217
2.3.11.3 设置挂机原因 218
2.3.11.3.1 SsmISDNSetHangupRzn 218
2.3.11.4 其它函数 219
2.3.11.4.1 SsmISDNGetDisplayMsg 219
2.3.11.4.2 SsmISDNGetTxCallerSubAddr 219
2.4 有关信号音发生器的函数（CTI系列） 220
2.4.1 设置信号音发生器的工作参数 220
2.4.1.1 SsmSetTxTonePara 220
2.4.1.2 SsmQueryOpSendTone 220
2.4.1.3 SsmGetTxTonePara 221
2.4.2 发送信号音 222
2.4.2.1 SsmSendTone 222
2.4.2.2 SsmSendToneEx 222
2.4.2.3 SsmStopSendTone 222
2.4.2.4 SsmChkSendTone 223
2.5 有关信号音检测器的函数 224
2.5.1 有关信号音检测器的常用函数 224
2.5.1.1 设置工作状态 224
2.5.1.1.1 SsmStartToneAnalyze 224
2.5.1.1.2 SsmCloseToneAnalyze 224
2.5.1.1.3 SsmQueryOpToneAnalyze 225
2.5.1.2 获取检测结果 225
2.5.1.2.1 SsmGetToneAnalyzeResult 225
2.5.1.3 清除检测结果 226
2.5.1.3.1 SsmClearToneAnalyzeResult 226
2.5.2 有关FFT的函数 227
2.5.2.1 设置工作参数 227
2.5.2.1.1 SsmSetPeakFrqDetectBW 227
2.5.2.2 获取工作参数 227
2.5.2.2.1 SsmQueryOpPeakFrqDetect 227
2.5.2.2.2 SsmGetPeakFrqDetectBW 228
2.5.2.3 获取检测结果 228
2.5.2.3.1 SsmGetPeakFrq 228
2.5.2.3.2 SsmGetPeakFrqEnergy 229
2.5.2.3.3 SsmGetOverallEnergy 229
2.5.2.3.4 SsmGetOverallEnergyAllCh 229
2.5.3 设置噪声滤波器的参数 230
2.5.3.1 SsmSetMinVocDtrEnergy 230
2.5.3.2 SsmGetMinVocDtrEnergy 231
2.5.4 有关呼叫进程音检测器的函数 231
2.5.4.1 设置第 2 呼叫进程音检测器的工作状态 231
2.5.4.1.1 SsmStart2ndToneAnalyzer 231

2.5.4.1.2 SsmGet2ndToneAnalyzerState 232
2.5.4.2 获取第 2 呼叫进程音检测器的结果 232
2.5.4.2.1 SsmGet2ndToneAnalyzeResult 232
2.5.4.2.2 SsmClear2ndToneAnalyzeResult 233
2.5.4.3 设置频率检测器的参数 233
2.5.4.3.1 SsmSetTonePara 233
2.5.4.3.2 SsmSet2ndTonePara 233
2.5.4.3.3 SsmGetTonePara 234
2.5.4.3.4 SsmGet2ndTonePara 234
2.5.4.4 设置拨号音检测器的参数 235
2.5.4.4.1 SsmSetIsDialToneDtrTime 235
2.5.4.4.2 SsmSet2ndIsDialToneDtrTime 235
2.5.4.4.3 SsmGetIsDialToneDtrTime 236
2.5.4.4.4 SsmGet2ndIsDialToneDtrTime 236
2.5.4.5 设置回铃音检测器的参数 236
2.5.4.5.1 SsmSetRingEchoTonePara 236
2.5.4.5.2 SsmSet2ndRingEchoTonePara 236
2.5.4.5.3 SsmGetRingEchoToneTime 237
2.5.4.5.4 SsmGetRingEchoTonePara 238
2.5.4.5.5 SsmGet2ndRingEchoTonePara 238
2.5.4.6 设置忙音检测器的参数 238
2.5.4.6.1 SsmSetBusyTonePeriodEx 238
2.5.4.6.2 SsmSetBusyTonePeriod 238
2.5.4.6.3 SsmSet2ndBusyTonePeriod 238
2.5.4.6.4 SsmSetIsBusyToneDtrCnt 239
2.5.4.6.5 SsmSet2ndIsBusyToneDtrCnt 239
2.5.4.6.6 SsmGetBusyTonePeriodEx 240
2.5.4.6.7 SsmGetBusyTonePeriod 240
2.5.4.6.8 SsmGet2ndBusyTonePeriod 240
2.5.4.6.9 SsmGetIsBusyToneDtrCnt 241
2.5.4.6.10 SsmGet2ndIsBusyToneDtrCnt 241
2.5.4.7 获取检测结果 241
2.5.4.7.1 SsmGetBusyToneLen 241
2.5.4.7.2 SsmGet2ndBusyToneLen 241
2.5.4.7.3 SsmGetBusyToneCount 242
2.5.4.7.4 SsmGet2ndBusyToneCount 242
2.5.4.8 获取采用背靠背忙音信号检测算法的检测结果 243
2.5.4.8.1 SsmGetBusyToneEx 243
2.5.5 有关传真进程音检测器的函数 243
2.5.5.1 设置工作参数 243
2.5.5.1.1 SsmSetVoiceFxPara 243
2.5.5.1.2 SsmGetVoiceFxPara 244
2.5.5.2 获取检测结果 245
2.5.5.2.1 SsmGetVocFxFlag 245

2.6 有关DTMF检测器的函数 245
2.6.1 启动/关闭DTMF检测器 245
2.6.1.1 SsmEnableRxDtmf 245
2.6.2 设置回调函数 246
2.6.2.1 SsmSetRxDtmfHandler 246
2.6.3 获取DTMF字符 247
2.6.3.1 SsmClearRxDtmfBuf 247
2.6.3.2 SsmGetDtmfStr 247
2.6.3.3 SsmGetDtmfStrA 247
2.6.3.4 SsmGetRxDtmfLen 248
2.6.3.5 SsmGet1stDtmf 248
2.6.3.6 SsmGet1stDtmfClr 248
2.6.3.7 SsmGetLastDtmf 249
2.6.4 使用WaitDtmf任务 249
2.6.4.1 SsmSetWaitDtmf 249
2.6.4.2 SsmSetWaitDtmfEx 249
2.6.4.3 SsmSetWaitDtmfExA 249
2.6.4.4 SsmCancelWaitDtmf 250
2.6.4.5 SsmChkWaitDtmf 251
2.7 有关DTMF发生器的函数（CTI系列） 252
2.7.1 设置DTMF发生器的参数 252
2.7.1.1 SsmSetTxDtmfPara 252
2.7.1.2 SsmQueryTxDtmf 252
2.7.1.3 SsmGetTxDtmfPara 253
2.7.2 启动DTMF发生任务 253
2.7.2.1 SsmTxDtmf 253
2.7.2.2 SsmStopTxDtmf 254
2.7.2.3 SsmChkTxDtmf 255
2.8 有关Barge-in检测器的函数 255
2.8.1 设置Barge-in检测器的参数 255
2.8.1.1 SsmSetBargeInSens 255
2.8.1.2 SsmSetIsBargeInDtrmTime 256
2.8.1.3 SsmSetNoSoundDtrmTime 256
2.8.1.4 SsmGetBargeInSens 257
2.8.1.5 SsmGetIsBargeInDtrmTime 257
2.8.1.6 SsmGetNoSoundDtrmTime 258
2.8.2 获取Barge-in检测器的检测结果 258
2.8.2.1 SsmDetectBargeIn 258
2.8.2.2 SsmDetectNoSound 259
2.8.2.3 SsmGetNoSoundTime 259
2.9 有关放音操作的函数 260
2.9.1 设置放音数据的去向 260
2.9.1.1 SsmSetPlayDest 260
2.9.2 设置放音的音量 260

2.9.2.1 SsmSetPlayVolume 260
2.9.2.2 SsmSetPlayGain 260
2.9.3 设置播放任务的终止条件 261
2.9.3.1 SsmSetDtmfStopPlay 261
2.9.3.2 SsmSetDTMFStopPlayCharSet 262
2.9.3.3 SsmGetDtmfStopPlayFlag 262
2.9.3.4 SsmGetDTMFStopPlayCharSet 263
2.9.3.5 SsmSetBargeinStopPlay 263
2.9.3.6 SsmGetBargeinStopPlayFlag 264
2.9.3.7 SsmSetHangupStopPlayFlag 264
2.9.4 播放单个文件的专用函数 265
2.9.4.1 SsmPlayFile 265
2.9.4.2 SsmStopPlayFile 266
2.9.4.3 SsmPausePlay 267
2.9.4.4 SsmRestartPlay 267
2.9.4.5 SsmFastFwdPlay 268
2.9.4.6 SsmFastBwdPlay 268
2.9.4.7 SsmSetPlayTime 269
2.9.4.8 SsmSetPlayPrct 269
2.9.4.9 SsmGetDataBytesToPlay 270
2.9.4.10 SsmGetDataBytesPlayed 270
2.9.4.11 SsmGetPlayedTimeEx 271
2.9.4.12 SsmGetPlayingFileInfo 271
2.9.5 播放文件序列的函数 272
2.9.5.1 SsmClearFileList 272
2.9.5.2 SsmAddToFileList 272
2.9.5.3 SsmPlayFileList 273
2.9.5.4 SsmStopPlayFileList 274
2.9.6 播放单个内存的函数 275
2.9.6.1 SsmPlayMem 275
2.9.6.2 SsmStopPlayMem 276
2.9.6.3 SsmSetStopPlayOffset 276
2.9.6.4 SsmGetPlayOffset 277
2.9.7 播放指定内存序列的函数 277
2.9.7.1 SsmAddToPlayMemList 277
2.9.7.2 SsmClearPlayMemList 278
2.9.7.3 SsmPlayMemList 278
2.9.7.4 SsmStopPlayMemList 279
2.9.8 播放预加载语音的函数（CTI系列） 280
2.9.8.1 预加载语音信息的初始化函数 280
2.9.8.1.1 SsmSetMaxIdxSeg 280
2.9.8.1.2 SsmGetTotalIndexSeg 280
2.9.8.1.3 SsmLoadIndexData 281
2.9.8.1.4 SsmFreeIndexData 282

2.9.8.1.5 SsmLoadChIndexData 282
2.9.8.1.6 SsmFreeChIndexData 283
2.9.8.2 播放预加载语音的函数 284
2.9.8.2.1 SsmPlayIndexString 284
2.9.8.2.2 SsmPlayIndexList 284
2.9.8.2.3 SsmStopPlayIndex 285
2.9.9 播放乒乓缓存的函数（CTI系列） 285
2.9.9.1 SsmPlayMemBlock 285
2.9.9.2 SsmStopPlayMemBlock 287
2.9.10 终止播放任务的通用函数 287
2.9.10.1 SsmStopPlay 287
2.9.11 获取播放任务的相关信息 288
2.9.11.1 SsmQueryPlayFormat 288
2.9.11.2 SsmGetPlayType 288
2.9.11.3 SsmCheckPlay 289
2.9.11.4 SsmGetPlayedPercentage 290
2.9.11.5 SsmGetPlayedTime 290
2.10 有关录音操作的函数 291
2.10.1 设置录音操作的参数 291
2.10.1.1 设置信号源及其音量 291
2.10.1.1.1 SsmSetRecVolume 291
2.10.1.1.2 SsmSetRecMixer 291
2.10.1.1.3 SsmQueryOpRecMixer 292
2.10.1.1.4 SsmGetRecMixerState 293
2.10.1.1.5 DTRSetMixerVolume 293
2.10.1.1.6 DTRGetMixerVolume 294
2.10.1.1.7 SsmSetRecBack 295
2.10.1.1.8 SsmSetNoModuleChBusRec 295
2.10.1.2 设置录音任务的终止条件 296
2.10.1.2.1 SsmSetDTMFStopRecCharSet 296
2.10.1.2.2 SsmGetDTMFStopRecCharSet 297
2.10.1.2.3 SsmSetHangupStopRecFlag 297
2.10.1.3 设置预录音功能（REC系列） 298
2.10.1.3.1 SsmSetPrerecord 298
2.10.1.3.2 SsmGetPrerecordState 298
2.10.1.4 设置录音文件截尾功能 299
2.10.1.4.1 SsmSetTruncateTail 299
2.10.1.4.2 SsmGetTruncateTailTime 300
2.10.1.5 设置录音AGC 300
2.10.1.5.1 SsmSetRecAGC 300
2.10.1.5.2 SsmGetRecAGCSwitch 301
2.10.2 获取录音操作的信息 301
2.10.2.1 SsmQueryRecFormat 301
2.10.2.2 SsmGetRecType 302

2.10.2.3 SsmGetRecTime 303
2.10.2.4 SsmCheckRecord 303
2.10.3 录制到文件的函数 304
2.10.3.1 SsmRecToFile 304
2.10.3.2 SsmRecToFileA 304
2.10.3.3 SsmRecToFileB 304
2.10.3.4 SsmRecToFileEx 304
2.10.3.5 SpyRecToFile 304
2.10.3.6 SsmStopRecToFile 307
2.10.3.7 SpyStopRecToFile 307
2.10.3.8 SsmPauseRecToFile 308
2.10.3.9 SsmRestartRecToFile 308
2.10.3.10 SsmChkRecToFile 309
2.10.3.11 SsmGetDataBytesToRecord 309
2.10.4 录制到指定内存的函数 310
2.10.4.1 SsmRecToMem 310
2.10.4.2 SsmStopRecToMem 311
2.10.4.3 SsmGetRecOffset 311
2.10.5 录制到指定乒乓缓存的函数（CTI系列） 312
2.10.5.1 SsmRecordMemBlock 312
2.10.5.2 SsmStopRecordMemBlock 313
2.11 有关交换总线的函数 314
2.11.1 双向连接函数（CTI系列） 314
2.11.1.1 建立双向连接 314
2.11.1.1.1 SsmTalkWith 314
2.11.1.1.2 SsmTalkWithEx 314
2.11.1.2 拆除双向连接 315
2.11.1.2.1 SsmStopTalkWith 315
2.11.2 单向连接函数 315
2.11.2.1 建立单向连接 315
2.11.2.1.1 SsmListenTo 315
2.11.2.1.2 SsmListenToEx 315
2.11.2.1.3 SsmLinkFrom 315
2.11.2.1.4 SsmLinkFromEx 315
2.11.2.1.5 SsmLinkFromAllCh 315
2.11.2.2 拆除单向连接 318
2.11.2.2.1 SsmStopListenTo 318
2.11.2.2.2 SsmStopLinkFrom 318
2.11.2.2.3 SsmUnLinkFromAllCh 318
2.11.3 获取总线信息函数 318
2.11.3.1 SsmGetChBusInfo 318
2.11.4 高级总线操作函数（CTI系列） 319
2.11.4.1 强制拆除通道的所有总线连接 319
2.11.4.1.1 SsmClearChBusLink 319

2.11.4.2 通道到总线时隙的单向连接 320
2.11.4.2.1 SsmLinkToBus 320
2.11.4.2.2 SsmUnLinkToBus 321
2.11.4.3 总线时隙到通道的单向连接 321
2.11.4.3.1 SsmLinkFromBus 321
2.11.4.3.2 SsmLinkFromBusEx 321
2.11.4.3.3 SsmUnLinkFromBus 322
2.11.4.3.4 SsmListenToPlay 323
2.11.4.3.5 SsmUnListenToPlay 323
2.12 有关板载音频功率放大器的函数 324
2.12.1 SsmQueryOpPowerAmp 324
2.12.2 SsmSetPowerAmpVlm 324
2.12.3 SetVolume 324
2.13 有关板载扬声器的函数 325
2.13.1 SsmSetLine0OutTo 325
2.14 有关回波抵消器（Echo Canceller）的函数（CTI系列） 326
2.14.1 设置回波抵消器的工作状态 326
2.14.1.1 SsmQueryOpEchoCanceller 326
2.14.1.2 SsmSetEchoCanceller 326
2.14.1.3 SsmGetEchoCancellerState 327
2.14.2 设置回波抵消器的工作参数 327
2.14.2.1 SsmSetEchoCancelDelaySize 327
2.14.2.2 SsmGetEchoCancelDelaySize 328
2.15 有关电话会议的函数（CTI系列） 328
2.15.1 会议室初始化函数 328
2.15.1.1 SsmCreateConfGroup 328
2.15.1.2 SsmFreeConfGroup 329
2.15.2 会议室成员操作函数 330
2.15.2.1 SsmJoinConfGroup 330
2.15.2.2 SsmExitConfGroup 330
2.15.2.3 SsmSetListenVlmInConf 331
2.15.3 获取会议室信息的函数 331
2.15.3.1 SsmGetConfCfgInfo 331
2.15.3.2 SsmGetTotalConfGroup 332
2.15.3.3 SsmGetConfGrpCfgInfo 332
2.15.3.4 SsmGetConfGrpInfo 333
2.15.3.5 SsmGetConfGrpId 333
2.15.3.6 SsmValidateGrpId 334
2.15.4 获取会议室成员信息的函数 334
2.15.4.1 SsmGetConfGrpMmbrId 334
2.15.4.2 SsmGetConfGrpMmbrInfo 335
2.15.4.3 SsmGetConfChInfo 335
2.16 有关FSK收发器的函数（CTI系列） 336
2.16.1 FSK数据发送器 336

2.16.1.1 SsmSetFskPara 336
2.16.1.2 SsmGetFskPara 337
2.16.1.3 SsmTransFskData 337
2.16.1.4 SsmStartSendFSK 338
2.16.1.5 SsmCheckSendFsk 338
2.16.1.6 SsmStopSendFsk 339
2.16.2 FSK数据接收器 339
2.16.2.1 SsmStartRcvFSK 339
2.16.2.2 SsmStartRcvFSK_II 341
2.16.2.3 SsmCheckRcvFSK 342
2.16.2.4 SsmStopRcvFSK 343
2.16.2.5 SsmClearRcvFSKBuf 343
2.16.2.6 SsmGetRcvFSK 344
2.17 有关电压检测器的函数 344
2.17.1 忽略电压检测结果 344
2.17.1.1 SsmSetIgnoreLineVoltage 344
2.17.1.2 SsmGetIgnoreLineVoltage 345
2.17.2 设置检测摘机/挂机动作的门限电压 345
2.17.2.1 SsmSetDtrmLineVoltage 345
2.17.2.2 SsmGetDtrmLineVoltage 346
2.17.3 获取电压检测器的输出结果 346
2.17.3.1 SsmGetLineVoltage 346
2.17.4 有关极性反转检测器的函数 347
2.17.4.1 SsmGetPolarRvrsCount 347
2.17.5 有关铃流信号检测器的函数 348
2.17.5.1 SsmGetRingCount 348
2.17.5.2 SsmClearRingCount 348
2.18 有关数字中继线信息的函数 349
2.18.1 SsmGetMaxPcm 349
2.18.2 SsmGetPcmInfo 349
2.18.3 SsmGetPcmLinkStatus 350
2.18.4 SsmPcmTsToCh 351
2.18.5 SsmChToPcmTs 352
2.18.6 SsmSetPcmClockMode 352
2.18.7 SsmGetInboundLinkSet 353
2.19 SHT系列板卡专用函数（CTI系列） 353
2.19.1 设置联合模块工作模式 353
2.19.1.1 SsmSetUnimoduleState 353
2.19.2 模拟中继通道专用函数 354
2.19.2.1 在模拟线路上产生闪断信号 354
2.19.2.1.1 SsmTxFlash 354
2.19.2.1.2 SsmSetTxFlashCharTime 354
2.19.2.1.3 SsmGetTxFlashCharTime 355
2.19.2.1.4 SsmChkTxFlash 355

2.19.2.1.5 SsmTxFlashEx 356
2.19.2.2 通道状态的强制迁移 357
2.19.2.2.1 SsmSetChState 357
2.19.2.3 去话接续时检测被叫摘机的函数 357
2.19.2.3.1 SsmStartPickupAnalyze 357
2.19.2.3.2 SsmSetCalleeHookDetectP 358
2.19.2.3.3 SsmGetPickup 358
2.19.2.4 检测通道摘机执行情况的函数 359
2.19.2.4.1 SsmCheckActualPickup 359
2.19.3 坐席通道专用函数 360
2.19.3.1 向电话机发送振铃信号 360
2.19.3.1.1 SsmStartRing 360
2.19.3.1.2 SsmStartRingWithCIDStr 360
2.19.3.1.3 SsmStartRingWithFskCID 360
2.19.3.1.4 fPcm_ConvertFskCID 361
2.19.3.1.5 SsmStopRing 361
2.19.3.1.6 SsmCheckSendRing 362
2.19.3.2 检测到话机摘机后自动送拨号音 362
2.19.3.2.1 SsmSetASDT 362
2.19.3.2.2 SsmGetASDT 363
2.19.3.3 检测话机的闪断信号 364
2.19.3.3.1 SsmSetLocalFlashTime 364
2.19.3.3.2 SsmGetFlashCount 364
2.19.3.3.3 SsmClearFlashCount 365
2.19.3.4 检测话机的摘机/挂机动作 365
2.19.3.4.1 SsmGetHookState 365
2.19.3.5 在电话线上产生极性反转信号 366
2.19.3.5.1 SsmSetPolarState 366
2.19.3.5.2 SsmGetPolarState 366
2.19.4 有关远端摘机检测器的函数（SHT系列only） 367
2.19.4.1 有关普通远端摘机检测器的函数 367
2.19.4.1.1 设置工作参数 367
2.19.4.1.1.1 SsmSetVoiceOnDetermineTime 367
2.19.4.1.2 获取工作参数 368
2.19.4.1.2.1 SsmGetVoiceOnDetermineTime 368
2.20 SHD系列板卡的高级编程API（CTI系列） 368
2.20.1 通道的闭塞控制（CTI系列） 368
2.20.1.1 闭塞本端的函数 368
2.20.1.1.1 闭塞单个通道 368
2.20.1.1.1.1 SsmBlockLocalCh 368
2.20.1.1.1.2 SsmUnblockLocalCh 369
2.20.1.1.1.3 SsmQueryLocalChBlockState 370
2.20.1.1.2 闭塞整条数字中继线 370
2.20.1.1.2.1 SsmBlockLocalPCM 370

2.20.1.1.2.2 SsmUnblockLocalPCM 371
2.20.1.1.2.3 SsmQueryLocalPCMBlockState 371
2.20.1.2 闭塞对端的函数 372
2.20.1.2.1 闭塞单个通道 372
2.20.1.2.1.1 SsmQueryOpBlockRemoteCh 372
2.20.1.2.1.2 SsmBlockRemoteCh 373
2.20.1.2.1.3 SsmUnblockRemoteCh 373
2.20.1.2.1.4 SsmGetRemoteChBlockStatus 374
2.20.1.2.2 闭塞整条数字中继线 374
2.20.1.2.2.1 SsmBlockRemotePCM 374
2.20.1.2.2.2 SsmUnblockRemotePCM 375
2.20.1.2.2.3 SsmGetRemotePCMBlockStatus 376
2.20.2 基于SS7-MTP3 的高级编程API 377
2.20.2.1 SsmSendSs7Msu 377
2.20.2.2 SsmGetSs7Msu 378
2.20.2.3 SsmGetMtp3State 378
2.20.2.4 SsmGetMtp3StateEx 378
2.20.2.5 SsmGetMtp2Status 379
2.20.2.6 SsmSendSs7MsuEx 380
2.20.3 有关ISDN信令的高级函数 380
2.20.3.1 SsmGetIsdnMsu 380
2.20.3.2 SsmSendIsdnMsu 381
2.20.3.3 SsmCheckIsdnMsu 381
2.20.3.4 SsmISDNGetStatus 382
2.20.4 有关中国 1 号信令的高级函数☑ 382
2.20.4.1 控制ABCD信令比特 382
2.20.4.1.1 SsmGetCAS 382
2.20.4.1.2 SsmSendCAS 383
2.20.4.1.3 SsmGetSendingCAS 383
2.20.4.1.4 SsmSetSendCASFlag 384
2.20.4.1.5 SsmGetSendCASFlag 384
2.20.4.2 控制R2 信号收发器 385
2.20.4.2.1 SsmSetRxR2Mode 385
2.20.4.2.2 SsmGetR2 386
2.20.4.2.3 SsmSendR2 386
2.20.4.2.4 SsmSendR2Ex 387
2.20.4.2.5 SsmStopSendR2 387
2.20.4.2.6 SsmGetSendingR2 388
2.21 SHV系列板卡的专用函数（CTI系列） 389
2.21.1 SsmGetMaxVCh 389
2.21.2 SsmGetMaxFreeVCh 389
2.21.3 SsmBindVCh 390
2.21.4 SsmUnBindVCh 391
2.21.5 SsmSetVoiceEffect 391

2.21.6 SsmGetVoiceEffect 392
2.22 SHN系列板卡的专用函数（CTI系列） 392
2.22.1 SIP通道的专用编程接口 392
2.22.1.1 SsmIpGetUsingCodecType 392
2.22.1.2 SsmIpSetForwardNum 393
2.22.1.3 SsmIpInitiateTransfer 393
2.22.1.4 SsmIpGetMessageField 394
2.22.2 H.323 通道的专用编程接口 394
2.22.2.1 SsmIpGetUsingCodecType 394
2.23 ATP系列录音卡专用函数（REC系列） 395
2.23.1 麦克风通道的输入增益控制 395
2.23.1.1 SsmQueryOpMicGain 395
2.23.1.2 SsmSetMicGain 395
2.23.1.3 SsmGetMicGain 396
2.24 DTP系列板卡的专用函数（REC系列） 397
2.24.1 基于状态机编程模式的函数 397
2.24.1.1 获取基本信息的函数 397
2.24.1.1.1 SpyGetMaxCic 397
2.24.1.1.2 SpyChToCic 397
2.24.1.2 获取呼叫进展的函数 398
2.24.1.2.1 SpyGetState 398
2.24.1.3 获取呼叫方向的函数 399
2.24.1.3.1 SpyGetCallInCh 399
2.24.1.3.2 SpyGetCallOutCh 399
2.24.1.4 取得主叫方号码和被叫方号码的函数 399
2.24.1.4.1 SpyGetCallerId 399
2.24.1.4.2 SpyGetCalleeId 399
2.24.1.5 获取数字中继线状态的函数 400
2.24.1.5.1 SpyGetLinkstatus 400
2.24.1.6 获取SS7 信令呼叫状态机同步输出的原始信令消息 401
2.24.1.6.1 SsmGetSs7SpyMsu 401
2.25 DST系列录音卡的专用函数（REC系列） 401
2.25.1 基于状态机编程模式的函数 401
2.25.1.1 DTRGetLCDStr 401
2.25.1.2 DTRGetDKeyStr 402
2.25.2 高级函数 402
2.25.2.1 DTRGetLCDStrA 402
2.26 有关语音编码格式转换的函数 403
2.26.1 fPcm_Mp3ConvertALaw 403
2.26.2 fPcm_Mp3ConvertULaw 403
2.26.3 fPcm_AdpcmToAlaw 404
2.26.4 fPcm_AdpcmToGsm 404
2.26.5 fPcm_AdpcmToMp3 404
2.26.6 fPcm_ AlawToUlaw 405

2.26.7 fPcm_ UlawToAlaw 405
2.26.8 fPcm_MemAdpcmToALAW 405
2.26.9 fPcm_MemAdpcmToULAW 405
2.26.10 fPcm_AlawConvertPcm16 406
2.26.11 fPcm_ALawConvertPcm8 406
2.26.12 fPCM_AlawConvertGC8 406
2.26.13 fPcm_ALawConvertMp3 406
2.26.14 fPcm_ULawConvertMp3 407
2.26.15 fPcm_MemAlawToPcm8 407
2.26.16 fPcm_MemAlawToPcm16 407
2.26.17 fPcm_MemGSMToPcm16 408
2.26.18 fPcm_Pcm16ConvertAlaw 409
2.26.19 fPcm_MemPcm8ToAlaw 409
2.26.20 fPcm_MemAlawToUlaw 410
2.26.21 fPcm_MemUlawtoAlaw 410
2.26.22 fPcm_MemPcm16ToAlaw 410
2.26.23 fPcm_GC8Convert 411
2.26.24 fPcm_Vox6KTo8K 411
2.26.25 fPcm_Vox8KTo6K 411
2.26.26 fPcm_ULawConvertGSM 412
2.26.27 fPCM_Pcm16ConvertG729A 413
2.26.28 fPcm_GetLastErrMsg 413
2.26.29 fPcm_Close 414
2.27 有关传真操作的函数（CTI系列） 414
2.27.1 设置传真操作的工作参数 414
2.27.1.1 SsmFaxSetMaxSpeed 414
2.27.1.2 SsmFaxSetID 415
2.27.2 发送传真 415
2.27.2.1 SsmFaxStartSend 415
2.27.2.2 SsmFaxStartSendEx 415
2.27.2.3 SsmFaxSendMultiFile 416
2.27.2.4 SsmFaxSendMultiFileEx 416
2.27.2.5 SsmFaxAppendSend 417
2.27.3 接收传真 418
2.27.3.1 SsmFaxStartReceive 418
2.27.4 终止传真任务 418
2.27.4.1 SsmFaxStop 418
2.27.5 获取传真操作的信息 419
2.27.5.1 SsmFaxGetSpeed 419
2.27.5.2 SsmFaxCheckEnd 419
2.27.5.3 SsmFaxGetChStateMsg 420
2.27.5.4 SsmFaxGetPages 420
2.27.5.5 SsmFaxGetID 421
2.27.5.6 SsmFaxGetMode 421

2.27.5.7 SsmFaxGetAllBytes 422
2.27.5.8 SsmFaxGetSendBytes 423
2.27.5.9 SsmFaxGetRcvBytes 423
2.27.6 .tif文件的相关操作 423
2.27.6.1 fBmp_ValidateFaxFile 423
2.27.6.2 fBmp_SetHeaderFormat 424
2.27.6.3 fBmp_AddTxtToTif 425
2.27.6.4 fBmp_AddTxtToTif_Big 425
2.27.6.5 fBmp_UniteTif 426
2.27.6.6 fBmp_CutTifHeader 427
2.27.6.7 fBmp_GetFileAllPage 427

3 SynCti驱动程序的配置 428

3.1 系统配置文件ShConfig.ini 428
3.1.1 必须配置项 428
3.1.1.1 设置板卡总数 428
3.1.1.2 设置通道编号 428
3.1.1.3 设置板卡信息 429
3.1.1.4 SHD/DTP系列板卡 429
3.1.1.4.1 设置板卡复位功能 429
3.1.1.4.2 设置数字中继线的工作参数 429
3.1.1.4.3 设置PCM的逻辑编号 431
3.1.1.4.4 设置SpyPCM编号（DTP系列） 431
3.1.1.4.5 设置SpyCIC编号（DTP系列） 432
3.1.1.4.6 SS7 信令的必须配置项（DTP系列） 433
3.1.1.4.7 SS7 信令的必须配置项（SHD系列） 433
3.1.1.4.7.1 设置使用ISUP协议的数字中继线 433
3.1.1.4.8 ISDN信令必须配置项（SHD系列） 434
3.1.1.4.8.1 设置运行模式 434
3.1.1.4.8.2 设置使用ISDN信令(用户侧)的数字中继线 434
3.1.1.4.8.3 设置使用ISDN信令(网络侧)的数字中继线 435
3.1.1.4.9 SS1 信令的必须配置项（SHD系列） 435
3.1.1.5 SHN系列板卡 435
3.1.1.5.1 设置IP通道的工作参数 435
3.1.1.5.2 SIP信令的必须配置项（SHN系列） 436
3.1.1.5.3 H.323 信令的必须配置项（SHN系列） 436
3.1.1.6 DST系列板卡（REC系列 only） 436
3.1.2 常用配置项 437
3.1.2.1 设置SHD/DTP系列板卡的协议 437
3.1.2.2 设置驱动程序抛出的事件 437
3.1.2.3 设置事件的数据结构 437
3.1.2.4 设置驱动程序内部事件队列的深度 437
3.1.2.5 设置状态机的通用参数 438
3.1.2.6 设置数字线路上的语音编码格式 439
3.1.2.7 设置信号音检测器 439

3.1.2.7.1 设置数字卡的信号音检测开关 439
3.1.2.7.2 设置侦测的信号音参数 439
3.1.2.7.3 设置噪声滤波器的工作参数 440
3.1.2.7.4 设置脉宽滤波器的工作参数T 440
3.1.2.7.5 设置第 1 呼叫进程音检测器 440
3.1.2.7.5.1 设置频率检测器的工作参数 440
3.1.2.7.5.2 设置拨号音检测器的工作参数 440
3.1.2.7.5.3 设置回铃音检测器的工作参数 441
3.1.2.7.5.4 设置忙音检测器的工作参数 441
3.1.2.7.5.5 设置自定义信号音检测器的工作参数 441
3.1.2.7.6 设置第 2 呼叫进程音检测器 442
3.1.2.7.6.1 设置工作状态 442
3.1.2.7.6.2 设置频率检测器的工作参数 442
3.1.2.7.6.3 设置拨号音检测器的工作参数 442
3.1.2.7.6.4 设置回铃音检测器的工作参数 442
3.1.2.7.6.5 设置忙音检测器的工作参数 443
3.1.2.7.6.6 设置自定义信号音检测器的工作参数 443
3.1.2.7.7 设置传真进程音检测器 444
3.1.2.7.8 设置简易忙音检测器 444
3.1.2.7.8.1 设置工作状态 444
3.1.2.7.8.2 设置噪声滤波器的工作参数 444
3.1.2.7.9 设置回铃音检测方式 445
3.1.2.7.9.1 设置回铃音检测方式 445
3.1.2.7.9.2 设置检测回铃音高电平时允许的误差 445
3.1.2.7.9.3 设置检测回铃音低电平时允许的误差 445
3.1.2.8 设置信号音发生器(CTI系列) 445
3.1.2.9 设置自动语音识别 445
3.1.2.10 设置电话会议参数(CTI系列) 446
3.1.2.11 设置放音操作 447
3.1.2.11.1 设置播放任务的终止条件 447
3.1.2.11.2 设置驱动程序内部的放音缓冲区 448
3.1.2.11.3 设置文件序列播放任务的参数 448
3.1.2.11.4 设置预加载文件播放任务的参数 448
3.1.2.11.5 设置内存序列播放任务的参数 449
3.1.2.11.6 设置文件播放任务的参数 449
3.1.2.11.7 设置播放操作的音量 449
3.1.2.11.8 设置播放操作的编码格式 449
3.1.2.11.9 设置播放操作的其它参数 449
3.1.2.12 设置录音操作 450
3.1.2.12.1 设置录音任务的终止条件 450
3.1.2.12.2 设置录音信号源 450
3.1.2.12.3 设置文件录音的截尾功能 451
3.1.2.12.4 设置驱动程序内部的录音缓冲区 451
3.1.2.12.5 设置录音文件为wav格式 451

3.1.2.12.6 设置AGC的工作参数 451
3.1.2.13 设置录音/放音的共用参数 453
3.1.2.13.1 设置乒乓缓冲区的最小值 453
3.1.2.13.2 设置软件编码的GSM格式 453
3.1.2.14 设置DTMF检测器 453
3.1.2.14.1 启动/关闭DTMF检测器 453
3.1.2.14.2 设置DTMF检测器的缓冲区大小 454
3.1.2.14.3 设置DTMF滤波器的参数 454
3.1.2.14.4 设置DTMF脉宽滤波器的参数 455
3.1.2.15 设置DTMF发生器(CTI系列) 456
3.1.2.15.1 设置发送缓冲区的大小 456
3.1.2.15.2 设置DTMF发生器的参数 456
3.1.2.16 设置回波抵消器(CTI系列) 457
3.1.2.16.1 设置工作状态 457
3.1.2.16.2 设置工作参数 457
3.1.2.17 设置Barge-in检测器 (CTI系列) 457
3.1.2.17.1 设置Barge in 检测器的启动方式 457
3.1.2.17.2 设置Barge in 检测器的工作参数 457
3.1.2.18 设置TDM总线连接 459
3.1.2.19 设置模拟电话线的Caller ID检测器（SHT＋ATP系列） 459
3.1.2.19.1 设置Caller ID检测器的工作模式 459
3.1.2.19.2 设置FSK制式的工作参数 460
3.1.2.19.3 设置DTMF制式的工作参数 461
3.1.2.20 设置铃流信号检测器（SHT+ATP系列） 461
3.1.2.21 设置FSK数据收发器(CTI系列) 462
3.1.2.21.1 设置FSK数据发送器的工作参数 462
3.1.2.22 设置FSK数据接收器(CTI系列) 462
3.1.2.23 设置传真资源通道的工作参数 463
3.1.2.24 设置电压检测器 464
3.1.2.24.1 忽略电压检测器的检测结果 464
3.1.2.24.2 设置摘机/挂机判决条件 464
3.1.2.24.3 设置断线监测的判定条件 464
3.1.2.24.4 设置极性反转信号是否影响通道的状态迁移 465
3.1.2.25 设置USB录音盒/语音盒中的内置板载扬声器 465
3.1.2.26 SHT系列板卡常用配置项(CTI系列) 465
3.1.2.26.1 模拟中继通道专用配置项 465
3.1.2.26.1.1 设置去话呼叫的参数 465
3.1.2.26.2 设置闪断信号及保持时间 466
3.1.2.26.2.1 远端摘机检测器 466
3.1.2.26.2.1.1 设置普通的远端摘机检测器的工作参数 466
3.1.2.26.2.1.2 设置增强的远端摘机检测器 467
3.1.2.26.3 坐席通道专用配置项 467
3.1.2.26.4 传真通道专用配置项 469
3.1.2.26.5 联合模块的专用配置项 469

3.1.2.26.6 无业务模块通道的专用配置项 470
3.1.2.26.7 磁石通道 470
3.1.2.27 SHD系列板卡常用配置项(CTI系列) 470
3.1.2.27.1 设置来话呼叫的收号规则 470
3.1.2.27.2 设置“自动应答来话呼叫”功能 472
3.1.2.27.3 设置去话呼叫的主叫号码 473
3.1.2.27.4 连接到远端坐席接入模块 473
3.1.2.27.5 设置板卡工作模式 474
3.1.2.27.6 SS1 信令的高级设置 474
3.1.2.27.6.1 选择国家 474
3.1.2.27.6.2 设置接续过程的基本工作参数 475
3.1.2.27.6.3 设置状态机中通道的工作方式 476
3.1.2.27.6.4 设置中国 1 号信令状态机的参数 476
3.1.2.27.6.4.1 设置状态机中去话呼叫的参数 477
3.1.2.27.6.4.2 与Dialogic的SS1 通道对接 478
3.1.2.27.6.4.3 应用程序退出时是否向闭塞对端 478
3.1.2.27.6.4.4 输出调试信息 479
3.1.2.27.6.5 设置LineSide协议的参数 479
3.1.2.27.7 SS7 信令的高级设置 479
3.1.2.27.7.1 设置数字中继线第 16 时隙的属性 479
3.1.2.27.7.2 设置地址信号中备用值的对应字符 480
3.1.2.27.7.3 设置SS7 信令服务器的IP地址 480
3.1.2.27.7.4 设置本机的IP地址 481
3.1.2.27.7.5 SS7 信令服务器输出语音信道 481
3.1.2.27.7.6 设置数字中继线SS7 电路 481
3.1.2.27.7.7 TUP协议高级配置项 481
3.1.2.27.7.7.1 设置电路群消息的电路范围 481
3.1.2.27.7.7.2 来话呼叫：定制ACM消息 482
3.1.2.27.7.7.3 来话呼叫：定制GRQ消息 482
3.1.2.27.7.7.4 去话呼叫：定制IAI/IAM消息 483
3.1.2.27.7.7.5 去话呼叫：设置应答GRQ消息的模式 485
3.1.2.27.7.7.6 不使用SynCTI提供的TUP协议状态机 485
3.1.2.27.7.7.7 输出调试信息 485
3.1.2.27.7.8 ISUP协议高级配置项 486
3.1.2.27.7.8.1 来话呼叫：定制本端接受来话呼叫时应答的消息类型 486
3.1.2.27.7.8.2 来话呼叫：定制后向呼叫指示码 486
3.1.2.27.7.8.3 来话呼叫：定制REL消息 487
3.1.2.27.7.8.4 来话呼叫：设置其它参数 488
3.1.2.27.7.8.5 去话呼叫：定制IAM消息 488
3.1.2.27.7.8.6 去话呼叫：设置应答INF消息的模式 491
3.1.2.27.7.8.7 不使用SynCTI提供的ISUP协议状态机 491
3.1.2.27.7.9 SCCP协议的高级配置项 491
3.1.2.27.8 ISDN信令的高级设置 492
3.1.2.27.8.1 设置接续模式 492

3.1.2.27.8.2 设置用户侧/网络侧数字中继线的工作参数 492
3.1.2.27.8.3 设置去话呼叫的工作参数 493
3.1.2.27.8.4 设置来话呼叫的工作参数 496
3.1.2.27.8.5 设置定时器 497
3.1.2.27.8.6 设置其它参数 498
3.1.2.27.8.7 输出调试信息 498
3.1.2.27.9 设置DSP芯片的工作方式 499
3.1.2.28 SHN系列板卡常用配置项(CTI系列) 499
3.1.2.28.1 SIP信令的高级设置 499
3.1.2.28.1.1 设置SIP信令的工作参数 499
3.1.2.28.1.2 设置注册服务器 500
3.1.2.28.1.3 设置其他参数 500
3.1.2.28.2 H.323 信令的高级设置 501
3.1.2.28.2.1 设置H.323 信令的工作参数 501
3.1.2.28.2.2 设置H.323 服务器 502
3.1.2.28.2.3 设置其他参数 502
3.1.2.29 DST系列板卡常用配置项(REC系列) 502
3.1.2.30 ATP系列板卡常用配置项(REC系列) 502
3.1.2.30.1 设置输入端的增益 502
3.1.2.30.2 设置空闲通道的能量检测 503
3.1.2.30.3 设置预录音功能 503
3.1.2.30.4 检测脉冲方式的按键信号 504
3.1.2.30.5 设置闪断信号的最大保持时间 504
3.1.2.30.6 设置振铃信号计数器的清零保护时间 504
3.1.2.30.7 设置软交换功能 504
3.1.2.31 DTP系列板卡常用配置项(REC系列) 504
3.1.2.31.1 将ISDN消息保存到文件 504
3.1.2.31.2 SS7 信令状态机同步输出原始信令消息 504
3.1.3 高级配置项 505
3.1.3.1 驱动程序支持多个进程 505
3.1.3.2 输出API函数调用的调试信息 505
3.2 预加载语音配置文件ShIndex.ini（CTI系列） 506
3.2.1 [System]节 506
3.2.2 [SegNo=x]节 506
4 SS7 信令方式（CTI系列） 508
4.1 SS7 信令方式的基本概念 508
4.1.1 OPC和DPC 508
4.1.2 直连和准直连 508
4.1.3 信令链路和链路组 508
4.2 Synway板卡的SS7 信令应用系统 509
4.3 SS7 信令服务器的配置 510
4.3.1 设置信令服务器的信令点编码 510
4.3.2 设置信令服务器的IP地址 510
4.3.3 设置MTP3 的工作模式和工作参数 510

4.3.4 设置客户端的信息 511
4.3.5 设置信令链路的物理位置 512
4.3.6 设置信令链路组 512
4.3.7 设置DPC 513
4.3.8 设置用户层消息的目的点码 513
4.3.9 设置TUP/ISUP消息的分发路由 514
4.3.10 设置服务器的启动特性 514
4.3.11 设置服务器的显示特性 515
4.4 SS7 信令服务器的应用及配置实例 515
4.4.1 单OPC/单DPC 515
4.4.1.1 单机单卡的应用系统 515
4.4.1.2 单机多卡的应用系统 516
4.4.1.3 多机系统 517
4.4.1.4 使用专用信令服务器的多机系统 518
4.4.2 单OPC/多DPC 519
4.4.3 多OPC的SS7 应用系统 520
4.4.4 高可靠性的应用系统 520
4.4.5 为第三方板卡提供SS7 信令服务 523
4.4.6 准直连的应用系统 524
附录 1 ISDN通道释放原因值表 526

版权声明

本文档是杭州三汇信息工程有限公司（以后简称三汇公司）“Synway 板卡驱动软件”产品的组成部分，三汇公司拥有该软件以及本文档的一切版权，受中华人民共和国法律的保护。未经本公司书面授权，任何人不得复制、传播、摘抄、修改本文档的全部或部分内容。使用本文档，即视为接受后面的“软件授权协议”。

三汇公司保留对本文档进行修改而不另行通知之权利。

三汇公司对本文档进行了仔细校对，力求文档内容准确、可靠，但并不保证绝无错误。

请在使用本产品前，自行确定所使用的相关技术文件及规格为最新有效之版本。若因贵公司使用本公司之文件或产品，而需要第三方之产品、专利或者著作等与其配合时，则应由贵公司负责取得第三方同意及授权。关于上述同意及授权，非属本公司应为保证之责任。

注：本文中提到的 Windows，Windows NT，Windows 2000，Windows XP 等是微软的注册商标，Dialogic 是英特尔的注册商标。

SynCTI 驱动程序授权协议

1、三汇公司（以下简称本公司）拥有“本软件及所有附属产品、文件和相关文档”（以下简称本产品）的完全版权。任何单位和个人在使用前须获得本公司的授权。

2、本公司授权符合以下全部条件的单位和个人免费使用本产品：

A、本产品与通过合法销售渠道购买的本公司硬件产品配合使用时；

B、不私自向第三方传播本产品和本产品的任何一部分。

3、除符合第2条以外的其他单位和个人需要使用本产品时，必须获得本公司的书面授权。

4、获得授权的单位和个人不得转让其获得的授权。

5、使用本产品，即表明完全理解本协议的所有条款并全部接受。

版本修订记录

版本号	发布日期	修订内容
Ver2.0	2001.12	创建本文档。
Ver3.0.0.0	2003.07	重大修订。
Ver4.7.3.0	2006.12	重大修订。
Ver4.8.0.0	2007.07	重大修订。
Ver4.8.0.1	2007.08	无。
Ver4.9.0.0	2007.10	重大修订。
Ver5.0.0.0	2007.12	重大修订。

注：此处只记载针对文档本身的主要修订记录，由于 SynCTI 驱动程序升级而对文档进行修订的详细信息请参考对应版本的《驱动程序升级手册》。

前 言

欢迎使用 Synway 板卡。SynCTI 是 Synway 板卡的驱动程序，本文档是 SynCTI 驱动程序的帮助文件，主要阅读对象是使用 SynCTI 驱动程序的应用程序开发接口（API）开发应用软件的软件工程师。本文档还可以作为使用 Synway 产品构建 CTI 应用系统的安装维护技术人员、产品销售人员的参考书。

Synway 所有的板卡产品，都使用一个称为 SynCTI 的统一的驱动程序及开发平台作为硬件和应用软件之间的桥梁。SynCTI 为不同型号的硬件板卡提供统一的 API 接口，因此，基于某一种型号的 Synway 板卡开发的应用系统，可以很方便地扩展到各种不同的环境下。

本文档由四章组成，第 1 章讲述语音卡编程的基础知识，其中包括 Synway 板卡和 SynCTI 驱动程序的基本组成、系统架构，实现各种主要功能的操作方法和注意事项。阅读完这一章，读者可以知道 Synway 板卡能做什么，以及如何使用 API 来实现。对于首次使用 Synway 产品的读者，建议您在编程之前首先详细阅读这一章，当然，在阅读时可以跳过与您的应用程序无关的内容。例如，如果您的应用系统只是用来处理模拟电话线，那么您可以跳过数字中继线部分内容。

第 2 章详细描述 SynCTI 驱动程序的 API 函数、驱动程序抛出的事件，并对相关的数据结构进行了详细的描述。

第 3 章详细描述 SynCTI 驱动程序的全部配置项。本章的内容按照功能分类进行排列，每一功能的相关配置项按照分为必须配置项和常用/高级配置项，高级配置项通常用于支持少量特殊环境和特殊用户的需求。

第 4 章描述如何配置和实现使用 SS7 号信令的应用系统。

虽然 Synway 公司对本书进行了仔细的校对，但其中仍难免有错误和遗漏的地方，我们对由此给读者带来的不便深表歉意。同时，如果您能对此提出改进意见，我们将会非常感谢。

1 Synway 板卡编程基础

1.1 Synway 板卡的分类（CTI 系列）

Synway 板卡分为 2 大类：CTI 系列和 REC 系列。CTI 系列板卡主要作为通讯的一方进行电话呼叫，REC 系列为录音卡，主要用于高阻复接方式的并线录音。CTI 系列板卡和 REC 系列板卡的驱动程序都包含在 SynCTI 驱动程序中。

CTI 系列包括四个子系列：

子系列名称	描述
SHT 系列	采用模块化结构，模拟线路
SHD 系列	数字中继线（E1/T1/J1）
SHV 系列	变声卡
SHN 系列	VoIP 卡

各子系列所包含板卡型号的详细列表：

系列	总线形式	板卡型号	备注
SHT	USB	SHT-2A/USB	简称“USB 语音盒”
		SHT-4A/USB	
		SHT-2B/USB	
		SHT-4B/USB	
	PCI	SHT-8A/PCI	
		SHT-8B/PCI	
		SHT-8B/PCI/FAX	
		SHT-8C/PCI/FAX	
		SHT-8C/PCI/EC	
		SHT-16A-CT/PCI	
		SHT-16B-CT/PCI	
		SHT-16B-CT/PCI/FAX	
		SHT-16B-CT/PCI/MP3	
		SHT-16C-CT/PCI/FAX	
		SHT-120A-CT/PCI	已停产
	cPCI	SHT-16B-CT/cPCI	
		SHT-16B-CT/cPCI/MP3	
		SHT-16B-CT/cPCI/FAX	
		SHT-120A-CT/cPCI	已停产
SHD	PCI	SHD-30A-CT/PCI/SS1	支持 SS1
		SHD-30A-CT/PCI/ISDN	支持 SS1、ISDN
		SHD-30A-CT/PCI/SS7	支持 SS1、ISDN、SS7
		SHD-30B-CT/PCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-60A-CT/PCI/SS1	支持 SS1
		SHD-60A-CT/PCI/ISDN	支持 SS1、ISDN
		SHD-60A-CT/PCI/SS7	支持 SS1、ISDN、SS7
		SHD-60B-CT/PCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-120A-CT/PCI/SS1	支持 SS1
		SHD-120A-CT/PCI/ISDN	支持 SS1、ISDN
		SHD-120A-CT/PCI/SS7	支持 SS1、ISDN、SS7
		SHD-120D-CT/PCI	支持 ISDN、SS7
		SHD-120D-CT/PCI/EC	支持 ISDN、SS7

		SHD-240D-CT/PCI	支持 ISDN、SS7
		SHD-240D-CT/PCI/EC	支持 ISDN、SS7
		SHD-30C-CT/PCI	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60C-CT/PCI	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
	cPCI	SHD-30A-CT/cPCI/SS7	支持 SS1、ISDN、SS7
		SHD-30B-CT/cPCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-60A-CT/cPCI/SS7	支持 SS1、ISDN、SS7
		SHD-60B-CT/cPCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-120A-CT/cPCI/SS7	支持 SS1、ISDN、SS7
		SHD-240A-CT/cPCI/SS7	支持 SS7
		SHD-240S-CT/cPCI/SS7	支持 SS7
		SHD-480A-CT/cPCI/SS7	支持 SS7
		SHD-480S-CT/cPCI/SS7	支持 SS7
		SHD-240C-CT/cPCI	支持 ISDN、SS7
		SHD-240C-CT/cPCI/EC	支持 ISDN、SS7
		SHD-480C-CT/cPCI	支持 ISDN、SS7
		SHD-480C-CT/cPCI/EC	支持 ISDN、SS7
SHV	PCI	SHV-120A-CT/PCI	
		SHV-240A-CT/PCI	
		SHV-240A-CT/cPCI	
SHN	PCI	SHN-32A-CT/PCI	支持 SIP、H.323 协议

1.2 Synway 板卡的分类（REC 系列）

Synway 板卡分为 2 大类：CTI 系列和 REC 系列。CTI 系列板卡主要作为通讯的一方进行电话呼叫，REC 系列为录音卡，主要用于高阻复接方式的并线录音。CTI 系列板卡和 REC 系列板卡的驱动程序都包含在 SynCTI 驱动程序中。

REC 系列包括扩四个子系列：

子系列名称	描述
ATP 系列	采用模块化结构，适用于对模拟电话线路或其它类型模拟语音信号进行高阻复接录音
DTP 系列	适用于对数字中继线同时进行信令链路和语音信道的高阻复接录音。支持 SS1/ISDN/SS7
DST 系列	适用于对数字话机线路（2B+D）进行高阻复接录音
SHF 系列	用于对录制成声音文件的传真数据进行解码，从而获得实际的传真图像，通常作为录音系统的配套产品使用

各子系列所包含板卡型号的详细列表：

系列	总线形式	板卡型号	备注
ATP 系列	USB	SHT-2A/USB	简称“USB 录音盒”
		SHT-4A/USB	
		SHT-2B/USB	
		SHT-4B/USB	
	PCI	SHT-8A/PCI	
		SHT-8B/PCI	
		SHT-16A-CT/PCI	
		SHT-16B-CT/PCI	
		SHT-16B-CT/PCI/MP3	支持 MP3 格式的硬件编码
	cPCI	SHT-16B-CT/cPCI	
		SHT-16B-CT/cPCI/MP3	支持 MP3 格式的硬件编码
DTP 系列	PCI	SHD-30A-CT/PCI/FJ	板卡内置高阻接入电路
		SHD-60A-CT/PCI/FJ	板卡内置高阻接入电路
		SHD-30B-CT/PCI/FJ	板卡内置高阻接入电路

		SHD-60B-CT/PCI/FJ	板卡内置高阻接入电路
DST 系列	USB	SHR-2DA/USB	
	PCI	SHR-16DA-CT/PCI	支持 PCI-X 总线
	PCI	SHR-24DA-CT/PCI	支持 PCI-X 总线

1.3 SynCTI 支持的 CODEC

Synway 板卡支持的语音编码/解码格式(CODEC)格式及特性如下表所示：

CODEC	简称	采样率	码率	字节/帧	字节/秒	CODEC 引擎	编码值
PCM8 无符号	PCM8	8000 Hz	64 Kbps	1	8000	主机 CPU	1
16 位线性 PCM	PCM16	8000 Hz	128 Kbps	2	16000	主机 CPU	-2
A-law	A-law	8000 Hz	64 Kbps	1	8000	板载 DSP	6
μ-law	μ-law	8000 Hz	64 Kbps	1	8000	板载 DSP	7
IMA ADPCM	IMA ADPCM	8000 Hz	32 Kbps	256	4055	板载 DSP	17
VOX	VOX	8000 Hz	32 Kbps	1	4000	板载 DSP	23
MP3	MP3	8000 Hz	8 Kbps	72	1000	板载 DSP 主机 CPU	85
GSM 6.10	GSM	8000 Hz	13 Kbps	65	1625	主机 CPU	49
G.729A	G.729A	8000 Hz	8 Kbps	10	1000	板载 DSP	65411

注：
(1) 如果板卡型号中带有“/MP3”标记， CODEC 引擎使用板载 DSP 芯片；否则使用主机 CPU
(2) VOX 格式是指 Dialogic 的 ADPCM 编码格式。

1.3.1 编码格式转换的工具函数

ShPcmHandle.dll提供了语音编码格式转换的工具函数，这些工具函数在使用时，无须先调用SsmStartCti函数初始化SynCTI驱动程序，但使用结束后务必调用fPcm_Close函数以释放资源。语音编码格式转换的函数包括：

函数名称	功能概述
fPcm_Mp3ConvertALaw	将 Synway 板卡录制的 MP3 格式文件转换为 A 率格式文件
fPcm_Mp3ConvertULaw	将 MP3 格式文件转换为μ率格式文件
fPcm_AdpcmToAlaw	将 Synway 板卡录制的 IMA ADPCM 格式文件转换为 A 率格式文件
fPcm_AdpcmToGsm	将 IMA ADPCM 格式文件转换为 GSM 6.10 格式文件
fPcm_AdpcmToMp3	将 IMA ADPCM 格式文件转换为 MP3 格式文件
fPcm_MemAdpcmToALAW	将使用 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 A-law
fPcm_MemAdpcmToULAW	将使用 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成μ-law
fPcm_AlawConvertPcm16	将 A-law 格式文件转换为 16 Bit PCM 格式
fPcm_ALawConvertPcm8	将 A-law 格式文件转换为 8 Bit 无符号 PCM 格式文件
fPCM_AlawConvertGC8	将 A-law 格式文件转换为 G.729A 格式文件
fPcm_ALawConvertMp3	将 A-law 格式文件转换为 MP3 格式文件
fPcm_MemAlawToPcm8	将保存在缓冲区中的 A-law 格式数据转换为 8 Bit 无符号 PCM 格式
fPcm_MemAlawToPcm16	将保存在缓冲区中的 A-law 格式数据转换为 16 Bit 无符号 PCM 格式
fPcm_MemGSMToPcm16	将保存在缓冲区中的 GSM 格式数据转换成 16 Bit PCM 格式。
fPcm_Pcm16ConvertAlaw	将 16 Bit PCM 格式文件转换为 A-law 格式的文件
fPcm_MemPcm8ToAlaw	将保存在缓冲区中的 8 Bit 无符号 PCM 格式数据转换成 A-law 格式

fPcm_GC8Convert	将 G.729A 格式的文件转换成其它语音编码格式的文件
fPcm_Vox6kTo8k fPcm_Vox8KTo6K	实现 VOX 格式的语音文件在 6K 与 8K 采样率之间进行转换

1.3.2 板卡支持的 CODEC

Synway 板卡的放音操作和录音操作所支持的 CODEC 如下表所示：

Sieries	Board Model	PCM8		PCM16		A-law		μ-law		IMA ADPCM		VOX		MP3		GSM		G.729A	
		DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD
SHT	SHT-2A/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-4A/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-2B/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-4B/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-8A/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-8B/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-8B/PCI/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-8C/PCI/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-8C/PCI/EC	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16A-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/PCI/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16C-CT/PCI/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/PCI/MP3	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	√	★	★	★	—
	SHT-120A-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/cPCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/cPCI/MP3	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	√	★	★	★	—
	SHT-16B-CT/cPCI/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-120A-CT/cPCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
SHD	SHD-30A-CT/PCI/SS1	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30A-CT/PCI/ISDN	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30A-CT/PCI/SS7	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30B-CT/PCI/SS7/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60A-CT/PCI/SS1	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60A-CT/PCI/ISDN	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60A-CT/PCI/SS7	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60B-CT/PCI/SS7/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-120A-CT/PCI/SS1	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-120A-CT/PCI/ISDN	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-120A-CT/PCI/SS7	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-120D-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-120D-CT/PCI/EC	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-240D-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-240D-CT/PCI/EC	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30C-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60C-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30A-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30B-CT/cPCI/SS7/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60A-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60B-CT/cPCI/SS7/FAX	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-120A-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-240A-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHD-240S-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHD-480A-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHD-480S-CT/cPCI/SS7	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—

	SHD-240C-CT/cPCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-240C-CT/cPCI/EC	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-480C-CT/cPCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-480C-CT/cPCI/EC	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
SHN	SHN-32A-CT/PCI	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	√	√
ATP	SHT-2A/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-2B/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-4A/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-4B/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-8A/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-8B/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16A-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/PCI/MP3	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	√	★	★	★	—
	SHT-16B-CT/cPCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHT-16B-CT/cPCI/MP3	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	√	★	★	★	—
DTP	SHD-30A-CT/PCI/FJ	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-60A-CT/PCI/FJ	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	—
	SHD-30B-CT/PCI/FJ	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	√
	SHD-60B-CT/PCI/FJ	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	√
DST	SHR-16DA-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	√
	SHR-24DA-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	—	★	★	★	★	★	√
	SHR-2DA-CT/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	√

图例： COD：Coder（编码器） DEC：Decoder（解码器）
√： 硬件支持 —：不支持
☆：软件支持，由驱动程序通过软件算法实现 ★：软件支持，通过外挂 ACM 程序实现

1.4 SynCTI 驱动程序支持的操作系统

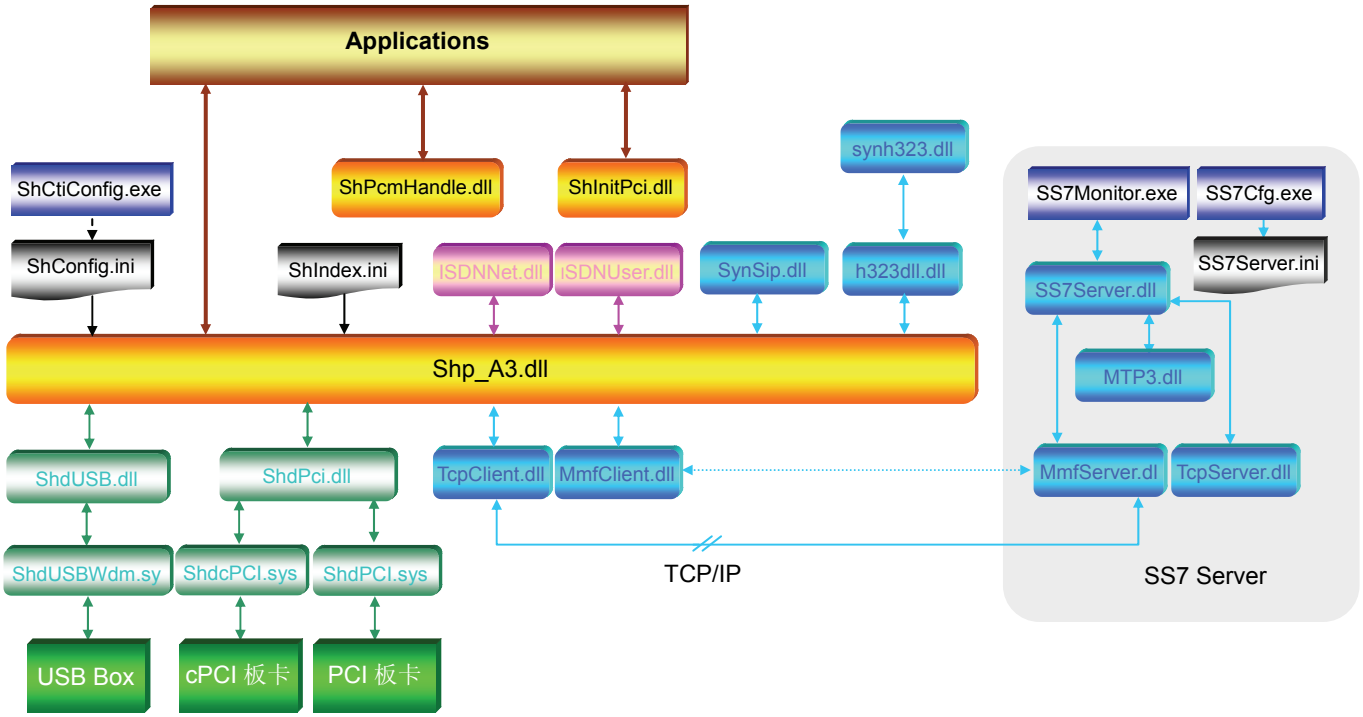
SynCTI 驱动程序支持 WINDOWS 和 LINUX 两大操作系统，具体如下：

- WINDOWS 操作系统：包括 Windows 98、Windows Me、Windows NT、Windows 2000、Windows XP
- LINUX 操作系统：包括 CentOS、Debian、FC4、FC6、FC7、Gentoo、Red Hat A4 U4、Red Hat AS3、RH9.0、Suse、Tribox2.0、Ubuntu

注意：对于采用 Compact PCI 总线的语音卡，必须使用 Windows 2000 以上操作系统。

1.5 SynCTI 驱动程序的体系结构（CTI 系列）

WINDOWS 操作系统下，SynCTI 驱动程序的体系结构如下图所示：

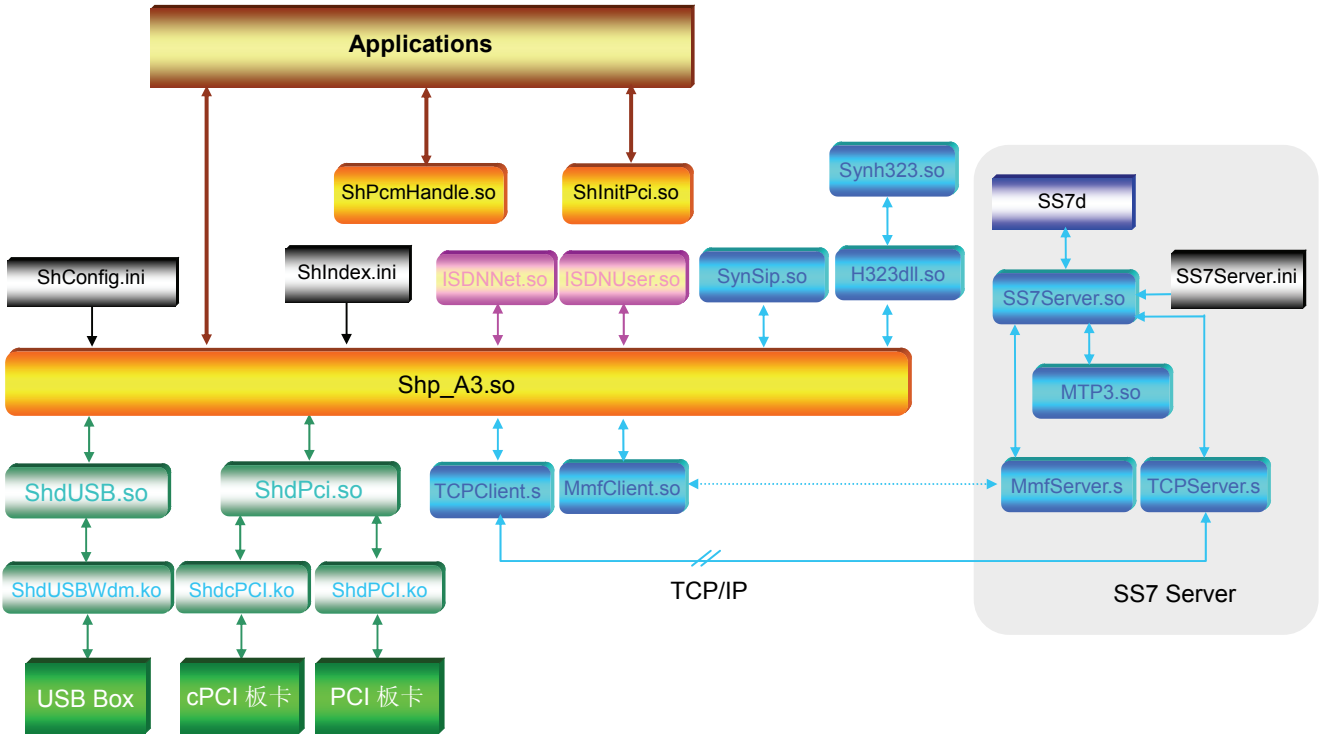


上图中，各模块的功能如下表所示：

模块名称	功能描述
Applications	用户应用程序
ShPcmHandle.dll	为应用程序提供 API 函数，实现各种音频格式的相互转换
ShInitPci.dll	为应用程序提供接口，便于用户获得板卡的基本信息
ShIndex.ini	内存索引语音对照表配置文件
ISDNNet.dll ISDNUser.dll	ISDN 消息处理模块。通过接口函数，对 A3 传递的命令或数据，依照协议进行分析和处理，再将所得结果通过接口函数传递给 A3；前者适用于网络侧模式，后者适用于用户侧模式
ShCtiConfig.exe	自动配置程序
ShConfig.ini	系统配置文件，与 ShCtiConfig.exe 相对应，详细配置信息参见驱动程序安装目录下的 ShConfig 高级选项样本.ini 文件
Shp_A3.dll	为应用程序提供接口 API 函数，帮助实现板卡的各种功能
ShdUSB.dll ShdPci.dll	在 A3 层与 SYS 层间进行数据、状态和命令的传递
ShdUSBWdm.sys ShdcPCI.sys ShdPCI.sys	系统 SYS 层。完成对硬件设备的添加、删除，负责和 DSP 程序进行通信等功能，三个模块分别处理 USB、cPCI、PCI 三种接口形式
SS7Monitor.exe	7 号信令服务器
SS7Cfg.exe	7 号信令配置程序
SS7Server.ini	7 号信令服务器配置文件
SS7Server.dll	7 号信令服务器调度模块，实现 SS7 的信令传递及路由控制等各项功能
MTP3.dll	7 号信令 MTP3 层功能模块
MmfServer.dll	7 号信令单机版服务器端模块，与 SS7Server.dll 及 MmfClient.dll 进行信令和数据的传递
MmfClient.dll	7 号信令单机版客户端模块，与 MmfServer.dll 及 A3 层进行信令和数据的传递
TcpServer.dll TcpClient.dll	7 号信令网络版服务器端模块及客户端模块，两模块间依据 TCP/IP 协议进行数据通信
USB Box、cPCI 板卡、PCI 板卡	接口分别为 USB、cPCI、PCI 形式的语音卡
SynSip.dll	SIP 信令处理模块

H323Dll.dll	专为处理 H.323 协议的软件适配层
SynH323.dll	H.323 消息处理模块

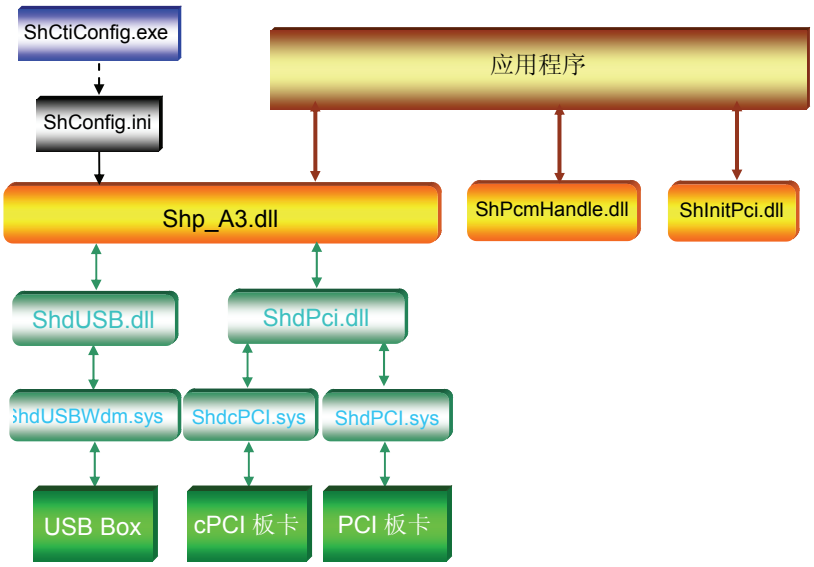
Linux 操作系统下，SynCTI 驱动程序的体系结构如下图所示：



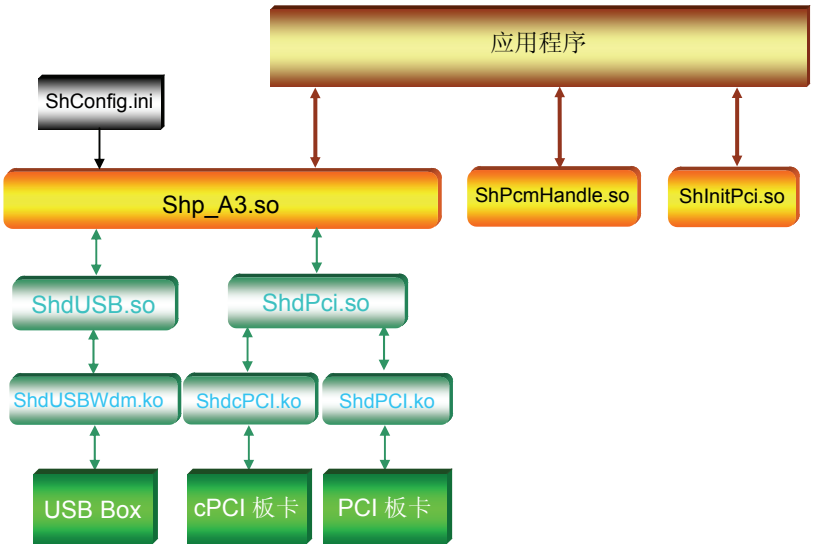
注：图中各模块的功能与 WINDOWS 操作系统下驱动程序体系结构图中的相应模块的功能一致。

1.6 SynCTI 驱动程序的体系结构（REC 系列）

WINDOWS 操作系统下，SynCTI 驱动程序的体系结构如下图所示：



Linux 操作系统下，SynCTI 驱动程序的体系结构如下图所示：



注：REC 系列驱动程序体系结构图中的各模块功能与 CTI 系列的相应模块的功能一致。

1.7 SynCTI 驱动程序的配套工具软件

1.7.1 系统配置工具软件 ShCtiConfig.exe

ShCtiConfig.exe 是对 ShConfig.INI 文件进行可视化配置的系统配置工具软件。ShCtiConfig.exe 支持对 Synway 所有型号板卡的常用参数配置。在完成 SynCTI 驱动程序的安装后，如果您不想通过手工编辑来取得有效的配置文件，只需运行 ShCtiConfig.exe，也可以进行参数设置。配置程序能够自动从板卡的固件中读取板卡的型号、序列号等信息，并可以自动计算通道数量等基本信息，并提供相应的控制按钮，将配置项的缺省值写入到 ShConfig.INI 文件中。

ShCtiConfig.exe 的主要功能包括：

- 对 PCI 板卡常用参数的可视化配置。
- 对 USB 语音盒常用参数的可视化配置。
- 对 cPCI 板卡常用参数的可视化配置和热插拔管理。

ShCtiConfig.exe 的具体使用方法请参考文档《系统配置工具软件用户手册》和 SynCTI 驱动程序安装目录下的 SS7Cfg_User_Manual.pdf 文件（英文）或 SS7Cfg_User_Manual_cn.pdf 文件（中文）。

1.7.2 信号音分析工具软件 ShTA.exe

在使用SHT系列、ATP系列板卡时，如果在模拟中继线上出现呼叫不成功、无法检测到对端挂机等情况，往往是由于呼叫进程音检测器的工作参数与交换机侧的参数不匹配所导致。某些小型交换机的信号音频率、周期与电信局用交换机的标准可能有差别，而Synway板卡上信号音检测器的工作参数是按照标准的局用交换机参数进行设置，因此，需要正确设置相关参数，如信号音的频率，拨号音、回铃音、忙音的占空比等，可以使用信号音分析工具软件ShTA.exe来对交换机侧的呼叫进程音的参数进行检测与分析，获得正确的设置值。

ShTA.exe 的具体使用方法请参考文档《信号音分析工具软件用户手册》，即 SynCTI 驱动程序安装目录下的 DemoShTAUserManual.pdf 文件（英文）或 DemoShTAUserManual_cn.pdf 文件（中文）。

1.7.3 SS7 信令服务器配置程序 SS7Cfg.exe

SS7Cfg.exe 是用于设置 SS7 信令服务器程序的配置文件 Ss7server.ini 的工具软件，具体的使用方法请参见文档《SS7 信令服务器配置程序用户手册》，即 SynCTI 驱动程序安装目录下的 SS7Cfg_User_Manual.pdf 文件（英文）或 SS7Cfg_User_Manual_cn.pdf 文件（中文）。

1.7.4 SS7 信令服务器 SS7monitor.exe

Ss7Monitor.exe 实现 SS7 信令的 MTP3 协议，同时具有信令消息分发和系统监视的作用。如果应用系统需要使用 SS7 信令，必须首先运行 SS7 信令服务器。有关 SS7 信令服务器的详细说明请参考《SS7 信令服务器用户手册》，即 SynCTI 驱动程序安装目录下的 SS7_Server_Manual.pdf 文件（英文）或 SS7_Server_Manual_cn.pdf 文件（中文）。

1.7.5 MSU 解码工具软件 MsuDecode.exe

MsuDecode.exe 是用来对 SS7 信令服务器的日志文件中的消息（MSU）进行翻译的工具软件。Ss7Monitor.exe 在运行时，可以根据用户的要求生成扩展名为.msu 的日志文件，将需要保存的消息以二进制的方式保存到文件中。MSU 解码工具软件可以将二进制形式的消息翻译成容易理解的文本形式的消息文件（.txt），可以直接进行阅读。有关 MSU 解码工具软件的具体使用方法请参考《MSU 解码工具软件用户手册》，即 SynCTI 驱动程序安装目录下的 MsuDecode_User_Manual.pdf 文件（英文）或 MsuDecode_User_Manual_cn.pdf 文件（中文）。

1.8 基本概念

1.8.1 通道（CTI 系列）

指板卡上用于处理一路电话的、具有完整的语音处理能力的物理电路实体。根据线路接口类型和信令的不同，通道的分类如下表所示：

板卡系列	业务模块 / 信令协议	通道类型	类型编码
SHT	外线模块	模拟中继线通道	0
	坐席模块	坐席通道	2
	录音模块	录音通道	3
	麦克风模块	录音通道	3
	磁石模块	磁石通道	10
	未安装任何业务模块	无模块通道	20
	传真资源通道	传真通道	5

SHD	SS1 信令	SS1 通道	4
	ISDN 信令（用户侧）	ISDN 通道（用户侧）	7
	ISDN 信令（网络侧）	ISDN 通道（网络侧）	8
	SS7 信令-TUP 协议	TUP 通道	6
	SS7 信令-ISUP 协议	ISUP 通道	11
	传真资源通道	传真通道	5
SHV		变声资源通道	14
SHN	H.323 信令	H.323 类型 IP 通道	15
	SIP 信令	SIP 类型 IP 通道	16

注：类型编码可以通过函数SsmGetChType获取。

1.8.2 通道（REC 系列）

指板卡上用于处理一路电话的、具有完整的语音处理能力的物理电路实体。根据线路接口类型和信令的不同，通道的分类如下表所示：

板卡系列	业务模块 / 信令协议	通道类型	类型编码
ATP	录音模块	模拟中继线录音通道	3
	麦克风模块	麦克风录音通道	3
DST		数字电话录音通道	12
DTP	SS1	SS1 监控通道	4
	ISDN	ISDN 监控通道	7 或 8
	SS7-TUP	TUP 监控通道	6
	SS7-ISUP	ISUP 监控通道	11

注：类型编码可以通过函数SsmGetChType获取。

1.8.3 通道的逻辑编号

驱动程序中，每个通道都有 2 个编号：物理编号和逻辑编号。

物理编号（简称 BCh）是指通道在所在板卡上的编号。通道物理编号由驱动程序自动分配，假设 N 表示板卡上的总通道数，M 表示总传真通道数，则板卡上通道的物理编号规则为：

板卡系列	通道的物理编号规则
SHT 系列	型号中没有“/FAX”标记：0~N-1 型号中包含“/FAX”标记：语音通道为 0~N-M-1，传真通道为 N-M~N-1
SHD 系列	0~N-1
SHV 系列	0~N-1
SHN 系列	0~N-1
ATP 系列	0~N-1
DTP 系列	CIC 编号：0~N-1 通道编号：0~N * 2 - 1
DST 系列	0~N-1

逻辑编号是指在应用系统中，为所有板卡上的全部通道进行统一编号。配置项TotalAppCh用于设置系统中的通道总数。本驱动程序中大部分函数使用的参数ch均指通道的逻辑编号。

通道的逻辑编号与物理编号之间是一一对应关系，由配置项AppCh决定。

注：在本手册中，如果不是特别说明，通道编号就是指通道的逻辑编号。

1.8.4 数字中继线

将用户终端设备（如用户交换机、集团电话、呼叫中心等）接入公用电话网络的电话交换机的线路称为中继线。

数字中继线是为用户提供数字线路联入电信网的中继线，通常为一对引自程控交换机的同轴电缆线或双绞线，一条用于发送，另一条用于接收。如果在电缆线上数据传输速率是 2.048 Mbps（简称 2M 线），中继线与用户终端设备之间的接口就称为 E1 接口；如果传输速率是 1.544Mbps，则称为 T1 接口。

数字中继线采用时分复用的工作方式，由多个时隙（Time Slot）组成，每个时隙的传输速率为 64 Kbps。E1 中继线具有 32 个时隙，T1 中继线具有 24 个时隙。

对于 E1 中继线，32 个时隙中的第 0 时隙被用作帧同步信息，因此通常只有 31 个有效时隙，其中，第 16 时隙通常被用于传输信令信息以及复帧同步信息，称为 D 信道；其余 30 个时隙被用作语音数据的传输时隙，称为 B 信道。

在本手册中，每一对 E1/T1 数字中继线均称为一条数字中继线。

1.8.4.1 数字中继线的帧同步

数字中继线的时隙 0 通常用来传递帧同步信息，因此不能作为语音时隙使用。SHD 系列和 DTP 系列的每条数字中继线端口都安装了帧同步指示灯，用于指示链路层的帧同步情况，灯亮表示帧同步正常，灯灭或闪烁表示帧同步失败。

如果使用 CAS 信令，复帧同步信息也在 0 时隙中传送。

当数字中继线的帧同步状态发生改变后，驱动程序会向应用程序抛出E_CHG_PcmLinkStatus事件。应用程序也可以随时调用函数SsmGetPcmLinkStatus获取数字中继线的当前状态。

1.8.4.2 数字中继线的时钟

每一条数字中继线都需要时钟才能正常工作，更多信息请参见本章中“配置应用系统的时钟”部分内容。

1.8.4.3 数字中继线的配置

本端的每条物理数字中继线必须与对端交换机的数字中继线的配置保持一致才能正常工作。

对数字中继线的设置是以板卡为基础，在**[BoardId=x]**节中进行设置。包括下列配置项：

```
[BoardId=x]
……
PcmNumber=M          //设置板卡内数字中继线的总数
PcmSSx[0]=1          //设置板卡内第 0#数字中继线：选择信令方式
PcmClockMode[0]=j    //选择时钟的工作模式
PcmLinkType[0]=k     //选择线路的物理传输类型
…….
```

1.8.4.4 数字中继线中的信令时隙

数字中继线中的信令时隙与协议类型有关，如下表所示：

协议类型	信令时隙
No.1 信令	第 16 时隙用来传送 ABCD 信令码
ISDN PRI	第 16 时隙用来传输信令消息
SS7 信令	通常将第 16 时隙用作信令链路。通过修改配置项UseTS16AsCircuit和Ss7SignalingTS也可以使用其它的时隙来承载信令链路。更多信息请参见本章中“时隙分配”部分内容。

1.8.4.5 数字中继线的编号

数字中继线有 2 个编号：物理编号和逻辑编号。

PCM 的物理编号是指该数字中继线在板卡内部的编号。板卡上的数字中继线从 0 开始编号，最大编号为板卡的数字中继线总数减 1。数字中继线的逻辑编号，是指在整个应用系统内，将所有板卡上数字中继线进行统一编制的号码。当应用程序内只有一片板卡时，数字中继线的逻辑编号与物理编号相同。

数字中继线的物理编号与逻辑编号之间的映射关系可以通过配置项TotalPcm和Pcm进行指定。

注意：在本手册中，如果不是特别说明，“PCM 编号”是指数字中继线的逻辑编号。

1.8.4.6 设置 B 信道的语音编码格式

配置项DefaultVoiceFormat用来设置数字用户线上的语音编码格式。

1.8.5 板卡的用户授权号

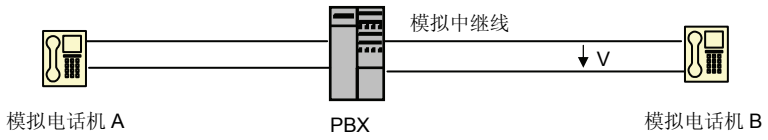
Synway 板卡上设计了具有加密功能的授权号电路，除了生产厂家之外，任何人不可能修改这一授权号。利用授权号功能，用户可以将应用程序与具体板卡绑定，相当于给用户软件装上了软件狗。使用了授权号后，您使用的 Synway 板卡就相当于您向三汇公司定做的一个特殊型号板卡。采用授权号保护用户的软件有以下优点：

- 操作简便：用户只要先向三汇公司申请一个授权号，然后由三汇公司将授权号写入提供给该用户的产品上，用户软件只要检查授权号是否符合即可，不需要任何算法，并且用户软件每次安装完毕不必作任何设置，不像采用系列号进行软件保护时需要进行复杂的加密计算，并且每次安装还要根据序列号进行一系列设置。
- 可靠性高：采用序列号进行软件保护时，如果加密算法外泄，则保护失效。而采用授权号方式，三汇公司在技术上保证除了生产厂家之外，没有任何人可以改写授权号，因此，对于申请了授权号的用户，即使授权号是公开的，其软件也绝对不可能运行在从其它渠道获得或者是三汇公司提供给其它用户的语音卡产品上。

函数SsmGetAccreditId或SsmGetAccreditIdEx可以用来获取板卡上的用户授权号。

1.8.6 模拟电话线的电压变化

模拟电话线上的直流工作电压为-48V，由交换机提供。Synway 的 SHT/ATP 系列板卡的每个通道上都具有电压检测器，用于检测模拟中继线上工作电压。下图是一个典型的电话连接示意图：



如果在模拟电话机 B 上进行如下操作：

（1） 在 t_1 时刻，B 摘机；

（2） 在 t_2 时刻，B 完成拨号，交换机向 A 振铃，向 B 送回铃音；

（3） 在 t_3 时刻，A 摘机，双方进入通话；

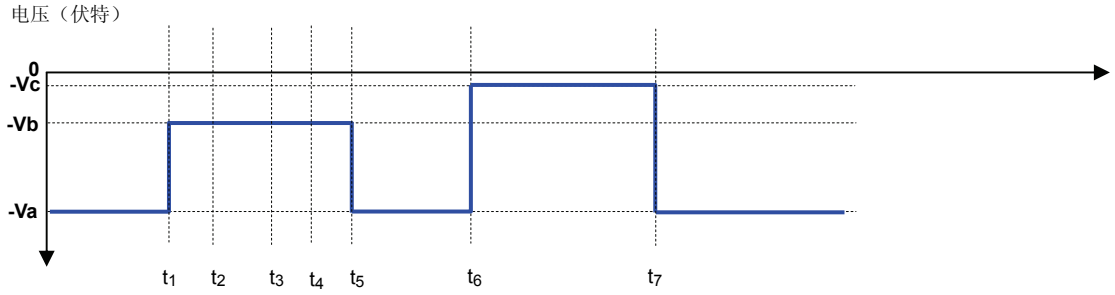
（4） 在 t_4 时刻，通话结束，A 挂机，交换机开始给 B 送忙音；

（5） 在 t_5 时刻，B 挂机；

（6） 在 t_6 时刻，B 与交换机的线路被断开；

（7） 在 t_7 时刻，B 与交换机的线路恢复连接。

如果交换机不支持极性反转功能，则连接模拟电话机 B 的线路上的电压 V 的变化情况如下图所示：



上图中，

Va：电话机处于挂机状态时的线路电压，一般为-48V；

Vb：电话机处于摘机状态时的线路电压，对于 Synway 板卡，一般为-15V 左右；

Vc：电话机的连线断开时，理论值为 0V；

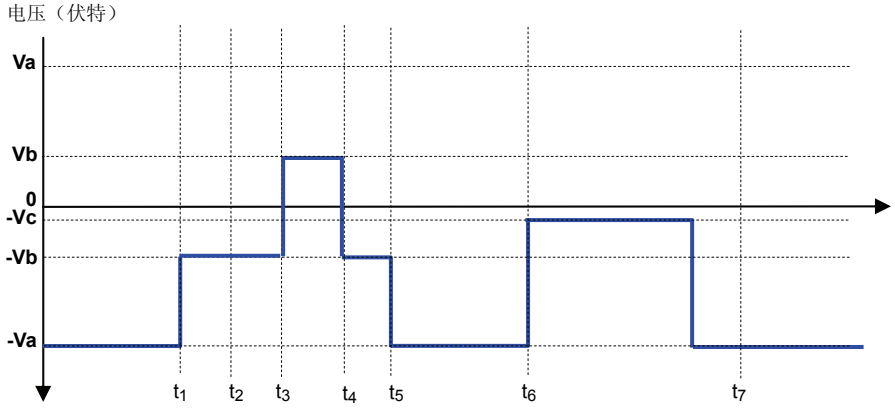
电压检测器利用线路上的实际电压值（假设为 Vin）与 Va、Vb、Vc 之间的关系来判断线路及话机的状态，判定标准可以表述为：

- ✧ 摘机：如果 Vc<Vin<Vb，并且持续一定时间 $t_{offhook}$；
- ✧ 挂机：如果 Vb≤Vin，并且持续一定时间 t_{onhook}；
- ✧ 断线：如果 Vin≤Vc，并且持续一定时间 t_{off}；

SynCTI 驱动程序中，用于设置上述参数的配置项或函数如下表所示：

参数名称	相关配置项或函数
Vc	配置项：OffLineSet
t_{off}	配置项：OffLineDetermineTime
Vb	配置项：IsHangupDtrmVoltage 函数：SsmSetDtrmLineVoltage
$t_{offhook}$	配置项：LineOncnt
t_{onhook}	配置项：MaxRecChFlashFilterTime

如果交换机支持极性反转功能，则连接模拟电话机 B 的线路上的电压 V 的变化情况如下图所示：



从上图可以看出，如果交换机支持极性反转功能，当被叫摘机或挂机时，交换机会变换主叫方的线路电压的极性，以向主叫方传递被叫的摘机/挂机信息。对于模拟线路，利用线路上电压的极性反转信号来判断被叫是否摘机是最准确的，但需要交换机的支持。

Synway的SHT和ATP系列板卡均内置了电压检测器。每当线路上的电压发生变化，并且变化的幅度超过预设值时，驱动程序会向应用程序抛出E_CHG_LineVoltage事件。电压变化的预设值为 5V，可以通过调用函数SsmSetEvent（携带E_CHG_LineVoltage参数）进行设置。应用程序也可以通过函数SsmGetLineVoltage获取线路的电压值。

电压检测器的作用：

➢ 监测本端与交换机端的线路连接是否正常。

SHT系列的模拟中继线通道、ATP系列的模拟中继线录音通道均支持断线监测功能。当线路电压低于配置项OffLineSet的设定值，并且保持时间大于配置项OffLineDetermineTime的设定值时，驱动程序会判定发生了断线故障，会将通道的状态迁移到“断线”状态。

➢ 判断线路上的摘机和挂机动作。

ATP 系列板卡的录音通道是根据线路电压(绝对值)在 Va 与 Vb 之间的变化情况，来判断话机是否摘机或挂机。

在缺省情况下，模拟录音通道会根据电压检测器的输出来判断话机的摘机/挂机状态，但通过修改函数SsmSetIgnoreLineVoltage或配置项IgnoreLineVoltage的设置，可以让驱动程序忽略电压检测器的检测结果，使录音通道一直保持在摘机状态。

对于麦克风录音模块，由于线路上没有馈电电压，因此，应该将配置项IgnoreLineVoltage设置为 1，这样可以使录音通道一直保持在“摘机”状态。

➢ 检测极性反转信号

模拟电话线上的馈电电压由交换机提供，通常为负极性，交换机可以改变馈电电压的极性（比如将-Vb 改变为 Vb），通过极性的变化向电话机传递一些信息。

利用模拟电话线的极性反转信号可以准确地判断被叫的摘机和挂机动作，但需要交换机的配合。ATP系列的模拟电话线录音通道和SHT系列的模拟中继线通道均支持极性反转检测功能。每当通道上检测到极性反转后，驱动程序会向应用程序抛出E_CHG_PolarRvrsCount事件，应用程序也可以通过函数SsmGetPolarRvrsCount获取极性反转信息。

对于模拟中继线通道，当检测到极性反转信号后，通道的状态是否发生迁移，取决于配置项DisablePolarReverse的设置。

1.8.7 模拟电话线上的铃流信号

当交换机通过模拟电话线呼叫话机时，会向话机发送铃流信号，使话机开始响铃。铃流信号通常是 1 秒 高电平、4 秒低电平的周期性信号。

ATP系列板卡的模拟中继线录音通道和SHT系列板卡的模拟中继线通道都具有铃流信号检测器。配置项RingDetectFilterPara用于设置铃流信号高电平和低电平的最小保持时间。每当铃流信号的电平发生变化时，驱动程序会向应用程序抛出E_CHG_RingFlag事件。

每当驱动程序检测到一次完整的铃流信号，都会向应用程序抛出E_CHG_RingCount事件。当铃流信号的累计个数到达指定值N时，通道状态会从“空闲”迁移到“振铃”。N可以通过配置项AlwaysToRingingOnRingCntX或ChToRingingOnRingCnt进行设置。实际的N按照下列规则计算：

（1） 如果没有在配置项ChToRingingOnRingCnt中指定或通过函数SsmSetFlag（携带F_ChToRingingOnRingCnt参数）设置，以配置项AlwaysToRingingOnRingCntX的设定值为准；

（2） 如果没有在配置项AlwaysToRingingOnRingCntX中指定，由驱动程序自动确定：如果Caller ID 检测器的工作模式设置为DTMF，N为 1，否则为 2；

应用程序也可以调用函数SsmGetRingCount来获取铃流信号的个数。

当满足下列条件时，驱动程序会自动将铃流信号的计数值清零：

✧ 通道从“振铃”状态迁移到“空闲”状态；

- 通道从“振铃”状态迁移到“摘机”状态，并且在“摘机”状态的保持时间超过配置项RecChClearRingDelayTime的设定值。

应用程序可以调用函数SsmClearRingCount将计数值清零。

1.8.8 模拟电话线上的闪断信号

闪断信号是指电话机进入通话后，通过快速拍打模拟电话机的叉簧而产生的短时挂机信号，通常用于与交换机配合实现转分机等操作。在话机上拍下叉簧的时间应该足够短，以免交换机将闪断信号误判为挂机信号。

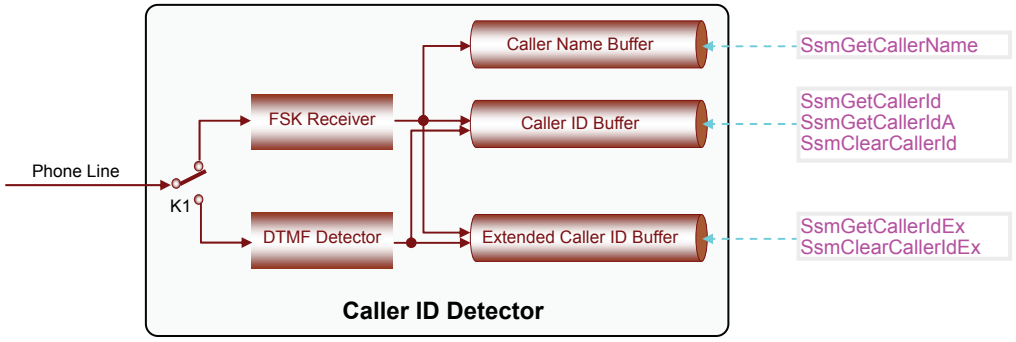
1.8.9 模拟电话线上的主叫号码

电话机的来电显示（Caller ID，主叫号码）是电信等运营商提供的一项付费业务，通常需要向其申请后才会获得这项功能。交换机向电话机传输主叫号码的方式有两种：

- FSK（频移键控）制式：Caller ID 是由交换机在给电话机的第一声振铃和第二声振铃之间进行传送；
- DTMF（双音多频）制式：在多数情况下，Caller ID 是由交换机在给电话机发送第一声振铃信号之前进行传送，但也有少数交换机在第一声振铃信号和第二声振铃信号之间向电话机发送。

一般的经验是，市局级交换机给模拟话机提供的来电显示号码是 FSK 制式传输的，小型交换机可能会以 DTMF 方式来传输。**注意：对于应用程序而言，如果需要完整地接收主叫号码，必须第二声铃声之后才能摘机。模拟外线通道在第几声振铃信号后才进入“振铃”状态，可以在配置文件中进行配置。**

Synway 的 SHT / ATP 系列板卡都具有 Caller ID 检测器，用于检测模拟电话线上的 FSK 制式/DTMF 制式的 Caller ID 信号，其原理框图如下图所示：



驱动程序内部为每个通道都设置了 2 个存放 Caller ID 的缓冲区：主叫号码缓冲区（Caller ID Buffer） 和主叫号码扩展缓冲区（Extended Caller ID Buffer）。主叫号码扩展缓冲区用于存放 FSK 制式的主叫号码的原始数据，里面包含了主叫号码、呼叫日期和时间、主叫方姓名等信息。驱动程序对原始信息进行下述处理：

- 分离出主叫方号码，并将其保存到主叫号码缓冲区中；
- 分离出主叫方姓名，并将其保存到主叫方姓名缓冲区中；

有关 Caller ID 检测器的函数、事件和配置项如下表所示：

类别	名称	描述
配置项	CallerIdStyle	设置 Caller ID 检测器的工作模式（FSK 制式 / DTMF 制式），即上图中的 k1 开关
	CloseCallerIdOnReceived FSKCallerIdDtrmTime FileterInvalidCID	设置 Caller ID 检测器工作在 FSK 制式时的工作参数

	DtmfCallerIDStyleLength DtmfCallerIDInterTimeout	设置 Caller ID 检测器工作在 DTMF 制式时的工作参数
函数	SsmSetFlag（携带F_CALLERIDSTYLE参数）	设置 Caller ID 检测器的工作模式
	SsmGetCallerId SsmGetCallerIdEx SsmGetCallerIdA	获取主叫方号码
	SsmGetCallerName	获取主叫方姓名
	SsmClearCallerId SsmClearCallerIdEx	清空驱动程序内部保存主叫号码的缓冲区
事件	E_CHG_CIDExBuf	Caller ID 检测器工作在 FSK 制式时，每当驱动程序完成一次主叫号码的接收后，会向应用程序抛出此事件

1.9 配置应用系统的时钟

每片 Synway 板卡都需要提供时钟信号才能正常工作。在设置基于 Synway 板卡的应用系统的时钟信号源时，需要考虑下列因素遵循下列原则：

- 如果板卡之间通过 CT-BUS 进行互联：
  - 只能由其中一片板卡提供唯一的时钟信号源，称为主时钟(Master Clock)，提供主时钟的板卡称为主卡(Master Board)，其它板卡称为从卡(Slave Board)，从卡的时钟信号通过 CT-BUS 从主卡获得，称为从时钟 (Slave Clock)；
  - 如果应用系统中至少包括一片 SHD 系列板卡，必须选择一片 SHD 系列板卡作为主卡；如果不包含任何 SHD 系列板卡，但包含 DST 系列或 DTP 系列板卡，应选择一片 DST 系列或 DTP 系列板卡作为主卡。
  - 如果主卡是 SHD 系列，主时钟信号只能由主卡上的一条数字中继线提供，主卡上的其余数字中继线必须工作在从时钟模式；
  - SHD 系列、DTP 系列板卡的时钟信号必须与公共电信网络保持一致，否则可能出现通话中有杂音、信令差错率高等不正常现象。
- 如果板卡没有通过 CT-BUS 与其它板卡进行互联，该卡必须设置为主卡；
- 通过 CT-BUS 与第三方板卡互联：如果将第三方板卡设置为主卡，需要将 Synway 板卡全部设置为从卡。

主卡提供的主时钟信号可以有 2 种来源方式：线路同步方式和自振荡方式。线路同步方式是指时钟信号提取自对端交换机连接到本端的输入信号，适用于 SHD 系列、DTP 系列和 DST 系列板卡，以保证与对端交换机的时钟信号保持同步；自振荡方式是指由板卡上内置的时钟振荡电路产生时钟信号，适用于 SHT 系列、ATP 系列板卡。如果使用 SHD 系列板卡来搭建一个测试系统，与实际的应用系统相连，则应该将测试系统的主卡的主时钟设置为自振荡模式。

SHV 系列板卡不能作为主卡，必须工作在从时钟模式。

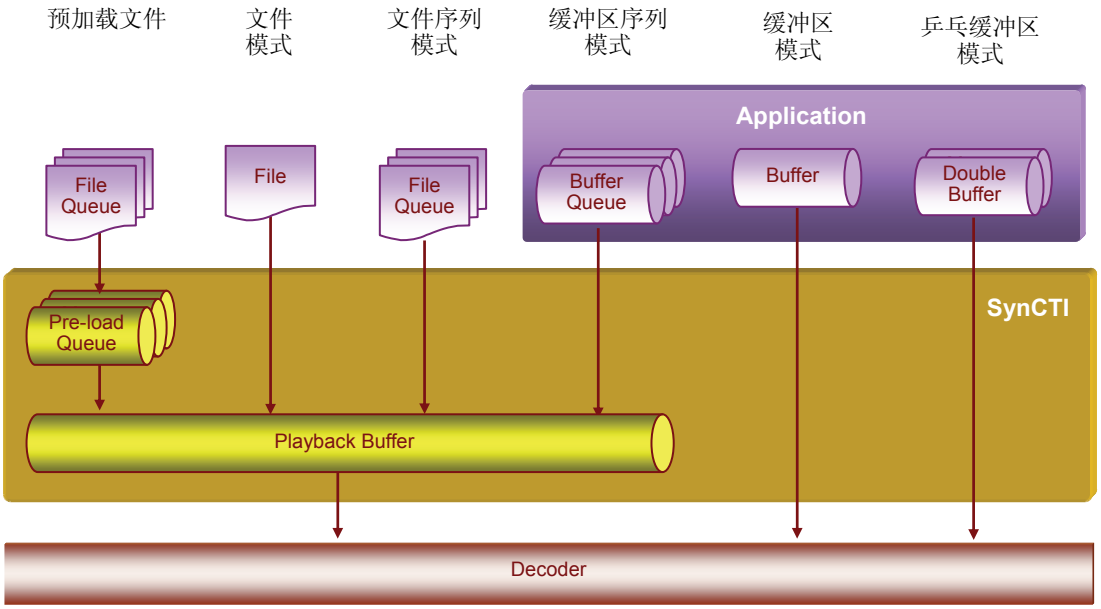
与时钟信号相关的配置项和函数如下表所示：

类别	名称	描述
配置项	WhoSupplySysClock	设置板卡的时钟模式（主卡/从卡）
	PcmClockMode	设置 SHD、DTP 系列板卡上数字中继线的时钟模式
函数	SsmSetPcmClockMode	设置 SHD、DTP 系列板卡上数字中继线的时钟模式，与配置项 PcmClockMode 的功能完全相同。

1.10 语音处理

1.10.1 语音播放（CTI 系列）

语音播放包括 6 种模式，如下图所示：



对于预加载语音片段、单个文件、文件序列、缓冲区序列这 4 种播放模式，放音操作启动后，驱动程序会先将源数据读取到内部的放音缓冲区(Playback Buffer)。放音缓冲区大小通过配置项PlayBufSize指定，缺省值为 32K字节。

1.10.1.1 设置放音数据音量及去向

1.10.1.1.1CTI 系列

有关放音数据的音量及其去向的详细内容，请参见本章中相应内容：

- SHD系列板卡：请参见本章中SHD系列板卡的工作原理框图部分；
- SHT系列板卡：请参见本章中SHT系列板卡的工作原理框图部分；
- SHN系列板卡：请参见本章中SHN系列板卡的工作原理框图部分。

1.10.1.1.2REC 系列

有关放音数据的音量及其去向的详细内容，请参见本章中相应内容：

- ATP系列板卡：请参见本章中ATP系列板卡的工作原理框图部分；
- DST系列板卡：请参见本章中DST系列板卡的工作原理框图部分；
- DTP系列板卡：请参见本章中DTP系列板卡的工作原理框图部分。

1.10.1.2 设置播放任务的终止条件

放音任务启动后，除了播放任务正常终止以及被应用程序调用停止播放的函数所终止外，如果通道上检测到了符合特定条件的事件，驱动程序也会自动终止放音任务，从而简化应用程序的编写。这些条件包括：

- DTMF 检测器在来话信号中检测到 DTMF 字符

相关函数及配置项包括：

函数名称	功能简述	相关配置项
SsmSetDtmfStopPlay	设置放音任务是否因DTMF检测器检测到DTMF字符而终止	DtmfStopPlayCharSet
SsmSetDTMFStopPlayCharSet	设置终止放音任务的 DTMF 字符集	
SsmGetDtmfStopPlayFlag	获取放音任务是否因 DTMF 检测器检测到 DTMF 字符而终止的标志	
SsmGetDTMFStopPlayCharSet	获取终止放音任务的 DTMF 字符集	

注意：当DTMF检测器在来话信号中检测到DTMF信号时，如果配置项DefaultPausePlayOnRxDtmf设置为 1，为了保证DTMF检测器的准确性，驱动程序会暂时停止语音的播放，待DTMF信号消失后，再自动恢复语音的播放。

- Barge in 检测器在线路上检测到了 Barge in（语音活动）

函数名称	功能简述	相关配置项
SsmSetBargeinStopPlay	设置放音任务是否因Barge in检测器检测到语音活动而终止	DefaultBargeInStopPlay
SsmGetBargeinStopPlayFlag	获取放音任务是否因Barge in检测器检测到语音活动而终止的标志	

- 呼叫状态机检测到对端挂机

函数名称	功能简述	相关配置项
SsmSetHangupStopPlayFlag	设置放音任务是否因驱动程序状态机检测到对端挂机而终止	HangupStopPlay

- 应用程序在通道上调用了会引发下总线操作的总线操作函数

SHD系列中某些型号板卡的特有属性，详细内容请参见本章中“SHD系列板卡的原理框图”部分。

1.10.1.3 播放预加载文件

对于IVR等应用，通常会需要应用程序播放一些常用的语音片段，例如提示语、合成日期、时间或数字等。为了提高应用系统的运行效率，降低应用程序访问硬盘的次数，SynCTI驱动程序提供了预加载文件模式，其工作原理为：事先将“0”、“1”、“2”、.....、“9”、“十”、“百”、“千”、“万”等常用语音片段录制到一个或多个文件中，并且为每条语音片段定义一个编号和别名，然后在调用驱动程序平台的初始化函数SsmStartCti时，由驱动程序自动将这些语音片段预先加载到内存，就可方便地播放诸如“125”等数字的连续语音。通过预加载文件模式，驱动程序可使语音片段实现“无缝连接”，效果大大优于使用一批单独的语音文件进行顺序播放。

驱动程序最多支持 65536 个预加载语音片段。

预加载语音片段可以通过编写配置文件ShIndex.ini实现。有关ShIndex.ini文件的详细信息请参见第 4 章的预加载语音配置文件ShIndex.ini。

应用程序也可以通过调用函数来动态加载语音片段。相关函数见下表所示：

- 有关加载和卸载语音片段的函数

函数名称	描述
SsmSetMaxIdxSeg	设置预加载语音片段的总数
SsmGetTotalIndexSeg	获取预加载语音片段的总数
SsmLoadIndexData	将指定的语音片段加载到内存
SsmFreeIndexData	卸载已经加载到内存的语音片段

- 有关播放预加载语音片段的专用函数

函数名称	描述
SsmPlayIndexList	顺序播放指定的一个或多个预加载语音片段
SsmPlayIndexString	顺序播放指定的一个或多个预加载语音片段
SsmStopPlayIndex	停止播放预加载语音片段

1.10.1.4 播放单个文件

播放单个文件，是指对单个语音文件进行播放。支持的文件格式包括标准的 WAV 文件和无格式文件（没有文件头的文件，编码格式需要另外指定）。

播放单个文件的相关函数及事件如下表所示：

函数或事件名称	描述
SsmPlayFile	播放单个文件，启动文件播放任务
SsmStopPlayFile	终止文件播放任务
SsmPausePlay	暂停（挂起）文件播放任务
SsmRestartPlay	恢复因为调用函数 SsmPausePlay 而暂停的文件播放任务
SsmFastFwdPlay	跳过 1 秒钟的语音数据后继续播放
SsmFastBwdPlay	回退 1 秒钟的语音数据后继续播放
SsmSetPlayTime	跳转到文件中的指定时间位置后继续播放
SsmSetPlayPrct	跳转到文件中的指定百分比位置后继续播放
SsmGetDataBytesToPlay	查询尚未播放的语音数据字节数
SsmGetDataBytesPlayed	查询已经完成播放的语音数据长度
SsmGetPlayingFileInfo	获取正在播放的放音文件的相关信息
SsmGetPlayedTimeEx	获得从文件首部的第 1 个语音数据字节开始，到文件中的当前放音位置的播放持续时间
E_PROC_PlayFile	文件播放任务的进展事件

1.10.1.5 播放文件序列

播放文件序列，是指应用程序连续提交多个具有相同语音编码格式的语音文件，由驱动程序负责按照提交的先后顺序，实现文件的连续自动播放，文件之间可以实现“无缝连接”。

函数SsmAddToFileList用来向驱动程序提交需要播放的文件。可以提交的文件数量的最大值由配置项MaxPlayFileList决定。

当应用程序调用SsmPlayFileList函数后，驱动程序内部就启动了一个文件序列播放任务，并开始播放第 1 个文件。每当完成一个文件的播放后，驱动程序会向应用程序抛出E_PROC_PlayFileList事件，然后开始播放下一个文件。当全部文件播放完毕后，驱动程序会自动关闭全部文件，终止文件序列播放任务，并向应用程序抛出E_PROC_PlayEnd事件。

播放文件序列的相关函数及输出事件如下表所示：

函数或事件名称	描述

SsmAddToFileList	向驱动程序提交一个需要播放的语音文件
SsmPlayFileList	启动文件序列播放任务
SsmClearFileList	关闭并清除已提交给驱动程序的全部语音文件
SsmStopPlayFileList	终止文件序列播放任务
E_PROC_PlayFileList	完成一个语音文件的播放后，驱动程序抛出的事件

1.10.1.6 播放单个内存

播放单个内存，是指由应用程序自行分配一块缓冲区，在将语音数据写入到缓冲区后，向驱动程序提交一个播放任务。驱动程序完成该缓冲区中的语音数据的播放后，向应用程序抛出事件，播放任务结束。

播放单个内存的相关函数及输出事件如下表所示：

函数或事件名称	描述
SsmPlayMem	启动单个内存播放任务
SsmSetStopPlayOffset	重新设置缓冲区中停止播放的位置
SsmStopPlayMem	终止单个内存播放任务
SsmGetPlayOffset	获取单个内存播放任务时，驱动程序内部的放音指针
E_PROC_PlayMem	输出单个内存播放任务的进展情况

1.10.1.7 播放内存序列

内存序列播放模式是指由应用程序以内存缓冲区的方式预先提交一组语音块，然后由驱动程序按照设定顺序对其进行连续播放的操作，语音块之间为无缝连接。语音块由应用程序自行管理。

播放内存序列的相关函数包括：

函数名称	描述
SsmAddToPlayMemList	向驱动程序提交一个内存缓冲区
SsmClearPlayMemList	清除已经提交给驱动程序的全部缓冲区
SsmPlayMemList	启动内存序列播放任务
SsmStopPlayMemList	终止内存序列播放任务

1.10.1.8 播放乒乓内存

乒乓内存播放模式采用双缓冲区机制。乒乓内存播放模式的工作机制为：在启动放音前，应用程序首先准备好一个（或两个）放音缓冲区并填充好放音数据，然后调用一次（或连续两次）SsmPlayMemBlock函数将缓冲区提交给驱动程序。驱动程序收到播放命令后，启动乒乓内存播放任务。当一个缓冲区内的语音数据播放完毕后，驱动程序自动播放另一个缓冲区，同时调用应用程序设置的回调函数，将完成播放的缓冲区信息传递给应用程序。应用程序在回调函数中将语音数据填充到下一个缓冲区，然后SsmPlayMemBlock函数再次将缓冲区提交给驱动程序。

与单个内存播放模式相比，乒乓内存播放模式具有下列特点：

- 可以将缓冲区最小设置为 192 字节；
- 支持回调的工作方式

由于 Windows 和 Linux 都不是真正的实时操作系统，将语音数据从应用程序的内存缓冲区传输到板卡的语音解码器，会产生一定的时间延迟。另外，由于语音数据需要先从用户态（应用程序）传输到内核态（设备驱动程序），再从内核态输出到板卡，所以这个时间延迟还不是恒定值，会随着应用系统的繁忙程度而变化。要减小传输延迟时间，比较有效的办法就是减小缓冲区的大小，同时提高应用程序更新语音数据的代码的优先级。驱动程序的线程优先级是非常高的，回调函数中的代码也就同时获得了与驱动程序相同的优先级。由此可见，乒乓内存播放模式可以将语音数据从应用程序的内存缓冲区转移到板卡的语音解码器所产生的时间延迟做到最小，适用于对放音操作要求实时性很高的应用场合。

播放乒乓内存的相关函数包括：

函数名称	描述
SsmPlayMemBlock	向驱动程序提交一个缓冲区，并启动乒乓内存播放任务
SsmStopPlayMemBlock	终止乒乓内存播放任务。

1.10.1.9 终止播放任务的通用函数

SsmStopPlay函数可以终止任何类型的播放任务。

1.10.1.10 获取播放任务的相关信息

下列函数可以用来获取播放任务的相关信息：

函数名称	描述
SsmQueryPlayFormat	查询通道是否支持指定的放音格式
SsmGetPlayType	查询通道上正在执行的放音任务类型
SsmCheckPlay	获取放音任务的完成情况
SsmGetPlayedPercentage	获取已经完成播放的数据长度占总长度的百分比
SsmGetPlayedTime	获取本次放音任务持续的时间

1.10.2 录音操作

1.10.2.1 设置录音信号源

录音操作的信号源与板卡的型号有关，详细内容请参见相应板卡系列的原理框图部分。

1.10.2.2 设置录音任务的终止条件

录音任务启动后，除了录音任务正常终止以及被应用程序调用停止录音的函数所终止外，如果通道上检测到了符合特定条件的事件，驱动程序也会自动终止录音任务，从而简化应用程序的编写。这些条件包括：

（1） DTMF检测器在来话信号中检测到DTMF字符

相关函数及配置项包括：

函数名称	功能简述	相关配置项
SsmSetDTMFStopRecCharSet	设置录音任务是否因DTMF检测器检测到DTMF字符而终止	DtmfStopRecCharSet

（2） 通道的呼叫状态机检测到对端挂机

相关函数及配置项包括：

函数名称	功能简述	相关配置项
SsmSetHangupStopRecFlag	设置录音任务是否因驱动程序状态机检测到对端挂机而终止	HangupStopRec

1.10.2.3 设置预录音功能

预录音是指由驱动程序按照预设参数启动一个预录音任务，将录音数据缓存到驱动程序内部的录音缓冲区中。当应用程序调用文件录音函数时，可以将缓冲区中的一段录音数据先保存到文件中，再开始正常的文件录音任务。预录音功能可以防止因应用程序响应迟缓等原因而导致前面语音数据的丢失。

预录音缓冲区的大小可以通过配置项RecordBufSize设置。

与预录音操作相关的函数或配置项包括：

函数名称	功能简述	相关配置项
SsmSetPrerecord	设置预录音功能。	PrerecordEnable PrerecordMode PrerecordInsertTime PrerecordCodec
SsmGetPrerecordState	获取预录音功能的设置	

1.10.2.4 设置录音文件截尾功能

录音文件截尾功能是指文件录音任务终止时，驱动程序自动将文件尾部的一段语音数据抛弃。此项功能适用于一些特殊的应用场合，例如，文件录音任务因为驱动程序检测到 DTMF 信号终止，但应用程序不希望该 DTMF 字符出现在录音文件中。

录音文件截尾功能只支持文件录音函数，并且必须与"检测到 DTMF 时自动终止文件录音任务"的功能配合使用。

与此项功能相关的函数或配置项包括：

函数名称	功能简述	相关配置项
SsmSetTruncateTail	设置录音文件截尾功能的工作参数	TruncateTailOnRecordToFile
SsmGetTruncateTailTime	获取预设的录音文件截尾功能的工作参数	

1.10.2.5 设置录音 AGC

自动增益控制（Automatic Gain Control,AGC）模块。如果开启 AGC 功能，驱动程序会根据输入信号的幅度对信号自动进行调节，对小信号进行提升，对大信号进行衰减。

函数SsmSetRecAGC用于开启或关闭AGC功能。AGC模块的工作参数可以通过下列配置项进行设置：

- AGCMAXGAIN
- AGCMINGAIN
- AGCMAXLEVEL
- AGCMINLEVEL
- AGCDOWNRATIO
- AGCUPRATIO
- AGCKEEPTIME

注意：

- **三汇语音卡的AGC的工作原理是根据原始输入信号的电平，自动调节AGC电路中的放大器增益系数，从而将输出信号的电平尽可能控制在预定的范围内。针对不同的应用环境和不同的用户偏好，可以通过设定AGC参数实现不同的AGC效果。三汇语音卡AGC的参数有 3 类：第一类，AGCMAXGAIN和AGCMINGAIN用于控制DSP增益系数的变化范围；第二类，AGCMAXLEVEL和AGCMINLEVEL用于设定期望的输出电平；第 3 类，AGCDOWNRATIO、AGCUPRATIO和AGCKEEPTIME用于控制DSP增益系数的变化速度。**
- **AGC 的缺省配置已经能够适应绝大多数的应用场景。如果确实需要调整 AGC 的工作参数，请在本公司的技术人员指导下进行。**
- **当通道迁移到S_CALL_STANDBY状态时，驱动程序会自动关闭AGC功能，应用程序仍然可以在通道上进行录音操作，但由于此状态下AGC功能已关闭，因此，录音数据不再具有AGC功能；若应用程序希望在这种情况下恢复录音数据的AGC功能，需要对配置项OpenRecEnAndPlayEnOnIdle进行设置。**

1.10.2.6 获取录音操作的信息

相关函数包括：

函数名称	功能简述
SsmQueryRecFormat	查询通道是否支持指定的录音格式

SsmGetRecType	获取正在执行的录音任务类型
SsmGetRecTime	获取录音任务持续的时间长度
SsmCheckRecord	查询录音任务的完成情况

1.10.2.7 录音函数概述

驱动程序在内部会为每一个通道分配一块录音缓冲区，缓冲区的大小可以通过配置项RecordBufSize进行设置，缺省值为 32768 字节。

录音函数包括文件录音、内存录音和乒乓缓冲区录音 3 类。

1.10.2.7.1 文件录音

相关函数包括：

函数名称	功能简述
SsmRecToFile SsmRecToFileA SsmRecToFileEx	启动文件录音任务
SsmStopRecToFile	终止文件录音任务
SsmPauseRecToFile	暂停文件录音任务
SsmRestartRecToFile	恢复文件录音任务
SsmChkRecToFile	查询文件录音任务的完成情况
SsmGetDataBytesToRecord	获取尚未完成的录音数据字节数

1.10.2.7.2 内存录音

内存录音是指应用程序调用SsmRecToMem向驱动程序提交一个缓冲区，驱动程序将录音数据写入该缓冲区，并按照设定的条件抛出进展消息。

相关函数包括：

函数名称	功能简述
SsmRecToMem	启动内存录音任务
SsmStopRecToMem	终止内存录音任务
SsmGetRecOffset	获取内存录音任务执行过程中，驱动程序内部的录音指针

1.10.2.7.3 乒乓缓冲区录音

乒乓缓冲区录音采用双缓冲区机制，它的工作原理为：在启动录音前，应用程序首先准备好一个（或两个）录音缓冲区，然后调用一次（或连续两次）SsmRecordMemBlock函数将缓冲区提交给驱动程序。应用程序首次调用SsmRecordMemBlock函数后，驱动程序会启动乒乓缓冲区录音任务，开始将录音数据写入到缓冲区。当一个缓冲区内写满后，驱动程序自动切换到另一个缓冲区，同时调用应用程序设置的回调函数，将完成录制的缓冲区信息传递给应用程序。在回调函数中，应用程序可以调用SsmRecordMemBlock函数提交新的缓冲区，并对录音数据进行后续处理。

与内存录音相比，乒乓缓冲区录音具有下列特点：

- 可以将缓冲区最小设置为 192 字节；
- 支持回调的工作方式

由于 Windows 和 Linux 都不是真正的实时操作系统，将录音数据从板卡传输到应用程序的内存缓冲区，会产生一定的时间延迟。另外，由于语音数据需要从内核态（设备驱动程序）传输到用户态（应用程序），涉及到从内核态到用户态的切换，因此这个时间延迟不是恒定值，会随着应用系统的繁忙程度而变化。要减小传输延迟时间，比较有效的办法就是减小缓冲区的大小，同时提高应用程序处理语音数据的代码的优先级。驱动程序的线程优先级非常高，回调函数中的代码具有与驱动程序相同的优先级。由此可见，乒乓缓冲区录音模式可以将语音数据的传输延迟减小到最低限度，适用于要求实时性很高的应用场合。

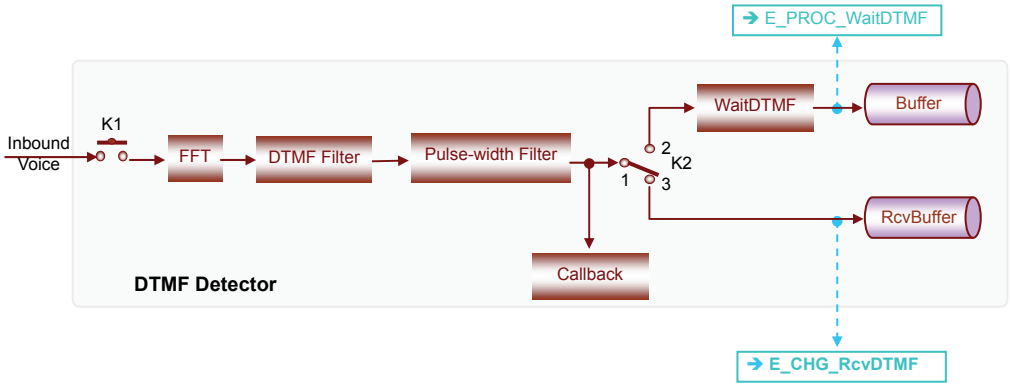
乒乓缓冲区录音的相关函数包括：

函数名称	功能简述
SsmRecordMemBlock	向驱动程序提交一个缓冲区，并启动乒乓内存录音任务
SsmStopRecordMemBlock	终止乒乓内存录音任务

1.10.3 DTMF 检测器(DTMF Detector)

三汇所有系列的板卡都为每个通道配置了独立的 DTMF 检测器。

DTMF 检测器的内部工作原理框图如下图所示：



开关K1 由配置项AlwaysEnableRxDtmf、驱动程序和应用程序共同控制。当AlwaysEnableRxDtmf设置为 1 时，K1 总是处于闭合状态；如果设置为 0，驱动程序会根据通道的状态迁移来决定K1 状态，当通道进入通话状态时闭合K1，离开通话状态并完成拆线后断开K1。应用程序可以调用SsmEnableRxDtmf来控制K1 的状态。

对于坐席通道，当通道处于“空闲”状态时，在驱动程序的自动控制下，DTMF检测器处于关闭状态，但可以通过配置项RcvDtmfOnIdle开启。

FFT 模块对输入语音信号进行 FFT 变换，将语音信号从时域变换到频域。

DTMF 滤波器（DTMF Filter）检查输入信号的频率特征中是否包含 DTMF 字符的信号，并以电平方式输出判断结果。

脉宽滤波器（Pulse-width Filter）用于对 DTMF 信号的脉冲宽度进行过滤，以消除语音信号中夹杂的 DTMF 信号对检测结果的影响。检测到的 DTMF 字符在送给 Callback 模块的同时，经由 K2 开关输出。

开关 K2 由驱动程序自动控制，用来控制 DTMF 的输出路径。K2 通常处于 1-3 连接状态，检测到的 DTMF 字符都送到 RcvBuffer 缓冲区中进行保存，同时向应用程序抛出 E_CHG_RcvDTMF 事件。

RcvBuffer缓冲区存放收到的DTMF字符，它的大小可通过配置项RxDtmfBufSize进行指定。访问RcvBuffer缓冲区的函数包括：

函数名称	功能简述
SsmClearRxDtmfBuf	清空驱动程序内部的 DTMF 按键号码接收缓冲区
SsmGetDtmfStr SsmGetDtmfStrA	获取保存在 DTMF 检测器缓冲区的 DTMF 字符
SsmGetRxDtmfLen	获取 DTMF 检测器缓冲区中保存的 DTMF 字符个数
SsmGet1stDtmf SsmGet1stDtmfClr	获取 DTMF 检测器缓冲区中最先收到的 DTMF 字符
SsmGetLastDtmf	获取 DTMF 检测器缓冲区中最后收到的 DTMF 字符

当应用程序调用SsmSetWaitDtmf函数启动一个WaitDtmf任务后，K2 自动切换到 1-2 连接状态，检测到的DTMF字符送到了WaitDtmf模块进行后续处理。注意：一旦启动了WaitDtmf任务以后，检测到的DTMF字符不再进入RcvBuffer中。

1.10.3.1 DTMF 滤波器（DTMF Filter）

DTMF 滤波器检查输入信号的频率特征中是否包含 DTMF 字符，并防止将语音信号误判为 DTMF 信号。

DTMF 字符与频率的对应关系如下表所示：

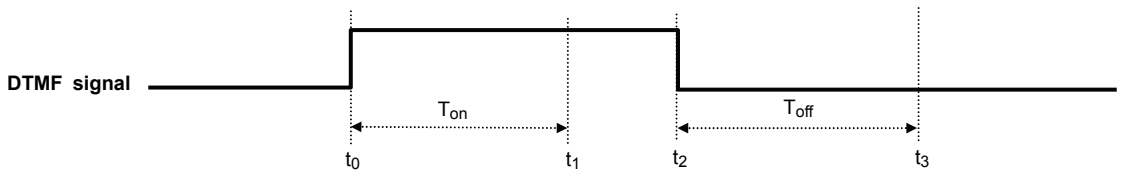
	1209Hz	1336Hz	1477Hz	1633Hz
697Hz	1	2	3	A
770Hz	4	5	6	B
852Hz	7	8	9	C
941Hz	*	0	#	D

下列配置项或函数用于强化 DTMF 滤波器的判决条件：

配置项或函数	描述
配置项：DualAndAllFreqEnScale 函数：SsmSetFlag（携带F_DualAndAllFreqEnScale参数）	设置 DTMF 信号的带内能量占全频能量的百分比判定门限。 DTMF 滤波器实时计算输入信号中 DTMF 的带内能量占全频能量的百分比，然后与某个设定的判定门限进行比较。如果大于设定门限，则判定线路上的信号为 DTMF 信号，并将判定结果以电平方式输出（高电平表示检测到信号音，低电平表示未检测到信号音）。
DtmfEnergy	设置 DTMF 信号的带内能量的绝对值。只有当 DTMF 信号的带内能量大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。
配置项：HighAndLowFreqEnScale 函数：SsmSetFlag（携带F_HighAndLowFreqEnScale 参数）	设置 DTMF 信号中，高频分量部分能量与低频分量部分能量之间的比例。只有当 DTMF 信号的高频分量部分能量与低频分量部分能量之间的比例大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。

1.10.3.2 DTMF 脉宽滤波器

DTMF 脉宽滤波器的工作原理如下图所示：



上图中，T_{on} 和 T_{off} 分别表示 DTMF 信号的高电平最小保持时间和低电平最小保持时间。当实际的输入 DTMF 信号电平从低电平转换到高电平后，只有当高电平保持时间和低电平保持时间分别大于 T_{on} 和 T_{off} 后，驱动程序才会判定线路上出现了真正的 DTMF 字符，并将此 DTMF 字符写入缓冲区，同时向应用程序抛出 E_CHG_RcvDTMF 事件。

T_{on}和T_{off}由配置项ReceiveDtmfSensitive或函数SsmSetFlag（携带F_RCVDTMFSENS参数）进行设置。

配置项OmitABCD用来决定是否丢弃收到的A、B、C或D字符。如果设置为保留，则收到的A、B、C或D字符将与其它字符一样进入后续处理模块，否则被丢弃。

配置项DtrmOnDtmfHighLevel用来决定驱动程序输出DTMF字符的时机。如果设置为高电平输出，当DTMF信号的高电平持续时间大于T_{on}时（即上图中的t_1 时刻）驱动程序就立即将检测到的DTMF字符输出；如果设置为低电平输出，则当高电平转换到低电平，并且低电平持续时间大于T_{off}时（即上图中的t_3 时刻），驱动程序才将检测到的DTMF字符输出。

1.10.3.3 WaitDtmf 任务

WaitDtmf任务用于向驱动程序提交一个接收特定DTMF字符串的任务。应用程序调用SsmSetWaitDtmf函数后，就启动了一个WaitDtmf任务。WaitDtmf任务启动后，DTMF检测器输出的DTMF字符不再进入RcvBuffer，而是直接进入了WaitDtmf模块的缓冲区，并根据预设条件判断是否应该终止本次任务。如果满足停止接收的条件，驱动程序向应用程序抛出E_PROC_WaitDTMF事件，本次WaitDtmf任务终止。

有关 WaitDtmf 的函数包括：

函数/事件名称	功能简述
SsmSetWaitDtmf SsmSetWaitDtmfEx SsmSetWaitDtmfExA	启动 WaitDtmf 任务
SsmCancelWaitDtmf	取消 WaitDtmf 任务
SsmChkWaitDtmf	查询 WaitDtmf 任务的进展情况
E_PROC_WaitDTMF	通知 WaitDtmf 任务的进展情况

1.10.3.4 Callback 方式编程

Callback模块处理应用程序的回调函数。当驱动程序检测到DTMF字符后，如果应用程序已通过调用函数SsmSetRxDtmfHandler设置了一个回调函数，会调用此回调函数。

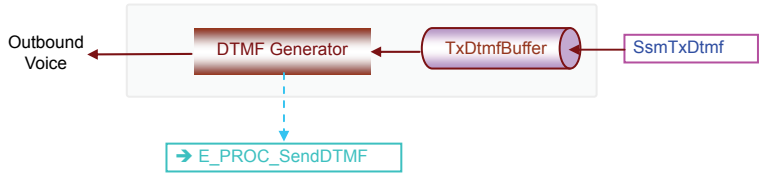
1.10.3.5 检测脉冲方式的按键信号

对于模拟中继线录音通道，DTMF检测器还支持脉冲方式的按键信息，但需要对配置项EnablePulseKeyDetect进行设置。

1.10.4 DTMF 发生器(DTMF Generator)（CTI 系列）

DTMF 发生器可以在线路上产生标准 DTMF 字符，包括：“0”、“1”、“2”、“3”、“4”、“5”、“6”、“7”、“8”、“9”、“*”、“#”、“A”、“B”、“C”、“D”。

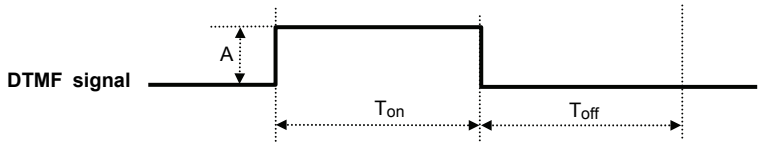
DTMF 发生器的原理框图如下图所示：



三汇板卡为每个通道都配置了独立的 DTMF 发生器（DTMF Generator）和发送缓冲区（TxDtmfBuffer）。

TxDtmfBuffer的大小可以在TxDtmfBufSize配置项中指定，缺省值为 50 个字符。

DTMF 发生器可以在线路上产生出频率、幅度完全符国家关于电话机 DTMF 拨号盘发号标准的 DTMF 信号。DTMF 发生器产生的典型 DTMF 信号如下图所示：



一个DTMF字符的波形特征用T_{on}和T_{off}这两个参数来表示，T_{on}表示高电平持续时间，T_{off}表示低电平持续时间，可以通过配置项TxDtmfTimePara或函数SsmSetTxDtmfPara进行设置。函数SsmGetTxDtmfPara用来获取T_{on}和T_{off}这两个参数的设置。

DTMF信号的幅度A可通过函数SsmSetFlag（携带F_TXDTMFAMP参数）设置。

应用程序调用SsmTxDtmf后，驱动程序会将待发送的DTMF字符存放在TxDtmfBuffer中，同时，启动一个发送DTMF的任务。每当DTMF发生器完成一个字符的发送，会自动从TxDtmfBuffer中取出下一个字符进行发送。如果TxDtmfBuffer为空，表明任务终止，驱动程序就向应用程序抛出E_PROC_SendDTMF事件。

函数SsmStopTxDtmf可以用来随时终止发送任务，SsmChkTxDtmf函数可以用来查询任务的进展情况。

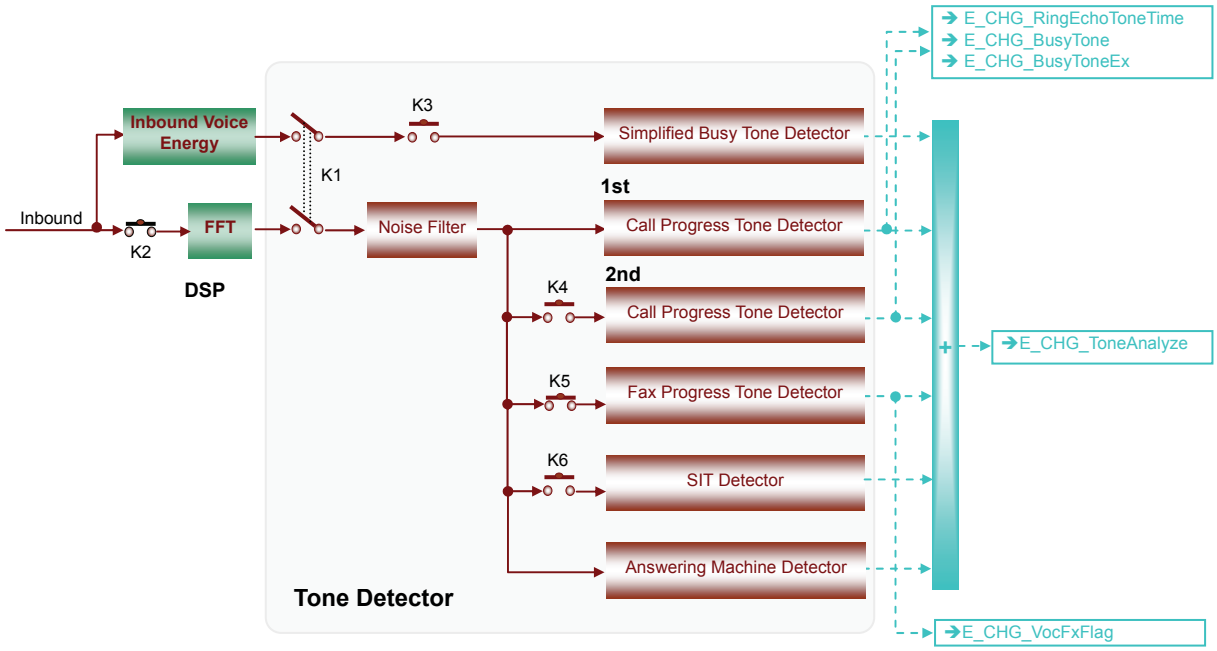
1.10.5 信号音检测器(Tone Detector)

每个三汇语音通道均内置了一个独立的信号音检测器。信号音检测器的主要作用：

- 模拟线路的呼叫进程音检测；
- 传真机信号音的检测
- 其它特殊应用，比如在自动呼叫测试系统中用于数字中继通道上的信号音检测等。

Tip：如果您的应用系统不包括上述类型的应用，可以不必阅读本小节。如果您的系统中包含模拟中继通道或模拟中继线录音通道，请仔细阅读本小节，本小节描述的内容对您应用系统的兼容性很重要。

信号音检测器的内部功能框图及相关事件输出如下图所示：



开关K1 控制信号音检测器的工作状态。K1 可以通过函数SsmStartToneAnalyze/ SsmCloseToneAnalyze来设置。注意：K1 开关同时控制了远端摘机检测器。驱动程序会根据通道的状态迁移情况，自动控制K1，系统初始化后，K1 处于关闭状态，详细内容请参见通道的状态迁移部分。

K2 由 DSP 程序自动控制，如果开启通道的 FSK 接收器，K2 断开，否则闭合。

整个信号音检测器由下列模块组成：

- **FFT 模块**

  FFT 模块把输入语音信号从时域变换到频域，变换结果送到后续模块进行信号音分析和计算。FFT 模块同时计算线路上语音信号的全频能量。FFT 的计算都在 DSP 芯片里完成，不占用主机资源。

- **噪声滤波器(Noise Filter)**

  用于过滤线路上的噪声。

- **呼叫进程音检测器（Call Progress Tone Detector）**

  用于检测线路上是否存在呼叫进程音。信号音的频率可以是单音频，也可以是双音频。每个信号音检测器包含了 2 个相互独立的呼叫进程音检测器。开关K4 用来控制是否启用第 2 呼叫进程音检测器，可以通过配置项Enable2ndToneAnalyzer或函数SsmStart2ndToneAnalyzer来设置，缺省设置为断开（关闭）。

- **传真进程音检测器（Fax Progress Tone Detector）**

  用于检测线路上是否存在单音频的传真进程音。每个信号音检测器包含了 2 个相互独立的传真进程音检测器，但当开关 K4 连接到第 2 呼叫进程音检测器时，传真进程音检测器不可用。

- **SIT 检测器（SIT Detector）**

  SIT（Special Information Tone）检测器，用于检测 SIT 信号音，常用于北美等地区。

- **机器应答检测器（Answering Machine Detector）**

  对端摘机后，机器应答检测器用于区分应答方是机器还是人。

- **简易忙音检测器(Simplified Busy Tone Detector)**

  对于模拟中继线通道，当应用程序启动了 FSK 接收器后，由于板卡上的 DSP 程序会自动断开 K2 开关，信号音检测器的输入被断开，无法正常工作。在某些使用 SHT 系列板卡的特殊应用场合，应用程序需要在接收 FSK 数据时仍然能够检测忙音，以及时判断对端的挂机动作。简易忙音检测器就是为了满足这种需要而提供的功能。当 K2 被 DSP 自动断开时，应用程序通过闭合 K1 和 K3，就可以使用简易忙音检测器。简易忙音检测器的输入只有 Barge-in 检测器输出的来话信号能量，因此无法判定忙音信号的频率特征，只能通过检测忙音的波形特征来判断线路上是否出现了忙音，因此称作简易忙音检测器。

1.10.5.1 FFT 模块

板载 DSP 芯片中的 FFT 模块对输入语音信号进行 FFT 变换，将语音信号从时域变换到频域，计算结果作为后续模块的输入。FFT 每隔 16 毫秒对线路上的语音信号进行一次 FFT 计算。

FFT 完成后，驱动程序会向应用程序抛出下列事件：

- E_OverallEnergy：将 FFT 计算得到的全频能量传递给应用程序。多数情况下应用程序都不必响应此事件。
- E_CHG_PeakFrq：将计算出的频谱中处于峰值位置的信号频率值及其带内能量传递给应用程序。函数SsmSetPeakFrqDetectBW用来设置峰值频率的带宽。多数情况下应用程序都不必响应此事件。**注意：只有当峰值频率值发生变化时才会抛出本事件。**

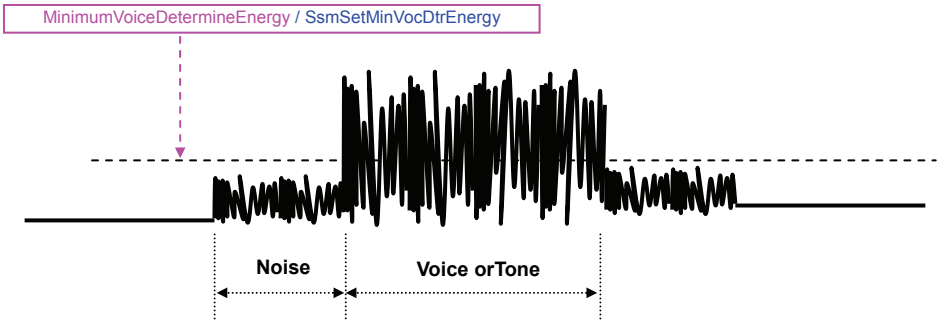
注意：

- **全频能量的输出所采用的量纲不是分贝（DB），这是为了降低计算量。如果应用程序需要使用 DB 作为量纲，则全频能量输出结果与 DB 的转换关系为（假设 X 为通过 API 函数获得的能量值）：**
  SHT 系列板卡： DB=10*lg（X/72,000,000）
  SHD 系列板卡： DB=10*lg（X/33,000,000）
  SHN 系列板卡： DB=10*lg（X/33,000,000）
- **如果启动了 FSK 接收器，FFT 模停止工作。**

1.10.5.2 噪声滤波器(Noise Filter)

在大多数情况下，噪声滤波器使用缺省配置即可，无须配置。

噪声滤波器模块的工作原理如下图所示：

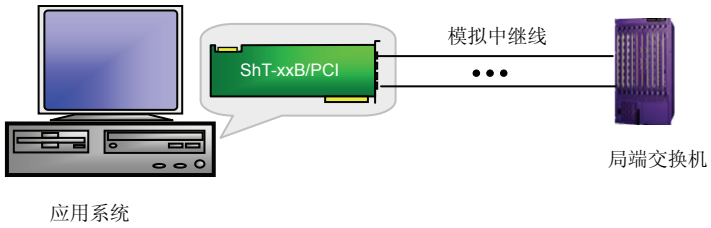


当全频能量小于配置项MinimumVoiceDetermineEnergy （或SsmSetMinVocDtrEnergy函数）指定的噪声门限，驱动程序会判定为噪声；否则判定为真实的语音信号。

1.10.5.3 呼叫进程音检测器(Call Progress Tone Detector) (SHT 系列 only)

对于模拟中继线通道，由于进程音检测器的分析结果直接影响到呼叫进程，因此，准确地使用和设置呼叫进程音显得非常重要。从应用系统的接入方式来看，通常有两种：直线接入方式和分机接入方式。

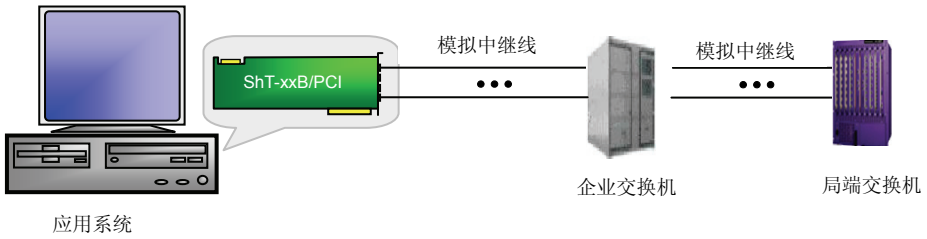
直线接入方式的应用系统结构如下图所示：



应用系统的直线接入方式

此类应用系统只需要检测一种类型的呼叫进程音。

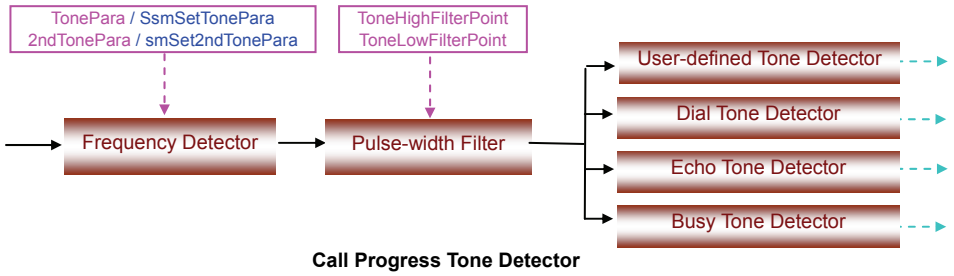
分机方式接入的应用系统结构如下图所示：



应用系统的分机接入方式

由于企业交换机、局端交换机的信号音可能不相同，因此，需要驱动程序能够同时检测 2 种不同的呼叫进程音。

呼叫进程音检测器由第 1 呼叫进程音检测器和第 2 呼叫进程音检测器组成，其工作原理框图如下图所示：



Call Progress Tone Detector

呼叫进程音检测器由频率检测器（Frequency Detector）、脉宽滤波器（Pulse-width Filter）、拨号音检测器（Dial Tone Detector）、回铃音检测器（Echo Tone Detector）、忙音检测器（Busy Tone Detector）和自定义信号音检测器（User-defined Tone Detector）组成。

自定义信号音检测器只用于某些特殊的场合。

拨号音检测器、回铃音检测器、忙音检测器和自定义信号音检测器可以同时工作。

- **频率检测器**

频率检测器检查输入信号的频率特征，判断是否包含指定频率的信号音。频率检测器实时计算输入信号中指定的信号音的带内能量占全频能量的百分比，然后与某个设定的判定门限进行比较。如果大于设定门限，则判定线路上的信号为信号音，并将判定结果以电平方式输出（高电平表示检测到信号音，低电平表示未检测到信号音）。

配置项TonePara或函数SsmSetTonePara用于设置第 1 呼叫进程音检测器的频率检测器的 2 个频率参数，包括中心频率和带宽，以及带内能量百分比判定门限。

配置项2ndTonePara或函数SsmSet2ndTonePara用于设置第 2 呼叫进程音检测器的频率检测器的频率参数，包括中心频率和带宽，以及带内能量百分比判定门限。

- **脉宽滤波器**

脉宽滤波器用于对信号音的脉冲宽度进行过滤，以消除语音信号中夹杂的信号音对检测结果的影响。当信号音电平从低电平向高电平跳变时，只有当高电平保持时间高于配置项ToneHighFilterPoint指定的时间，驱动程序才确认信号音的开始；当信号音电平从高电平向低电平跳变，只有当低电平保持时间高于配置项ToneLowFilterPoint指定的时间时，驱动程序才确认信号音的消失。

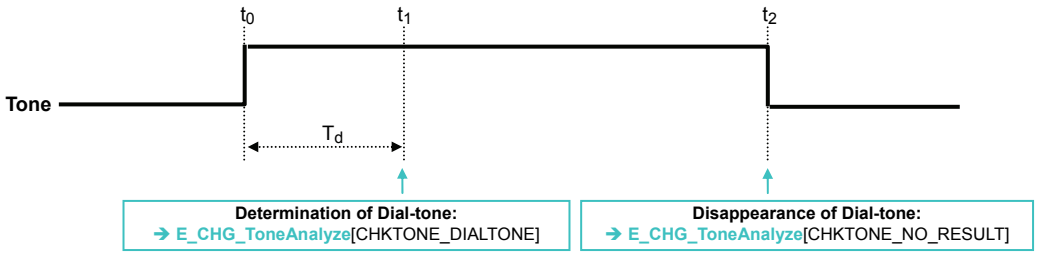
注意：

（1） 脉宽滤波器的配置参数同时对 1 呼叫进程音检测器、第 2 呼叫进程音检测器以及传真进程音检测器起作用。

（2） 对于模拟中继线类型的通道，第 1 呼叫进程音检测器中的拨号音检测器、回铃音检测器、忙音检测器的检测结果会自动作为状态机的输入事件；而对于第 2 呼叫进程音检测器，拨号音检测器、回铃音检测器、忙音检测器的检测结果是否作为状态机的输入事件，取决于配置项Check2ndToneOnAutoDial的设置。

1.10.5.3.1拨号音检测器（Dial Tone Detector）

拨号音的检测由拨号音检测器(Dial Tone Detector)完成。典型的拨号音波形如下图所示：

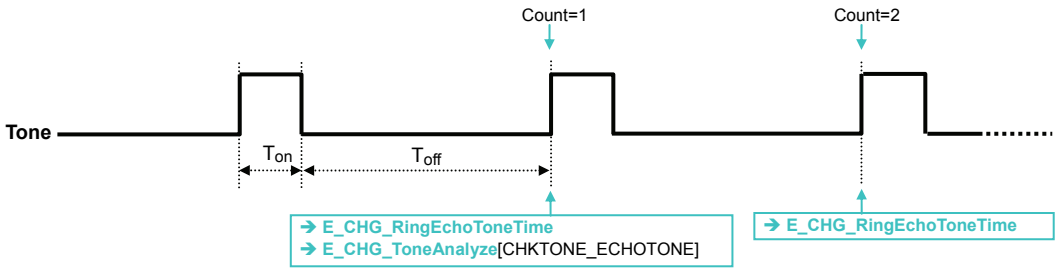


驱动程序对信号音的高电平进行计时，当计时到达拨号音判别门限 T_d 后，信号音检测器就会抛出 E_CHG_ToneAnalyze 事件（带 CHKTONE_DIALTONE 参数）。当拨号音消失后，拨号音检测器会抛出 E_CHG_ToneAnalyze 事件（带 CHKTONE_NO_RESULT 参数）。

对于第 1 呼叫进程音检测器的拨号音检测器，T_d 由配置项 IsDialToneDetermineTime 或函数 SsmSetIsDialToneDtrTime 进行指定；对于第 2 呼叫进程音检测器的拨号音检测器(如果开启)，T_d 由配置项 2ndIsDialToneDetermineTime 或函数 SsmSet2ndIsDialToneDtrTime 进行指定。

1.10.5.3.2回铃音检测器

回铃音的检测由回铃音检测器(Echo Tone Detector)完成。典型的回铃音波形如下图所示：



回铃音的主要参数包括：高电平持续时间 T_{on} 和低电平持续时间 T_{off}。当检测到一次与设定值相吻合的高电平时间和低电平时间，呼叫进程音分析器就判决为回铃音，回铃音计数器将加 1。T_{on} 和 T_{off} 允许的误差均为±20%。

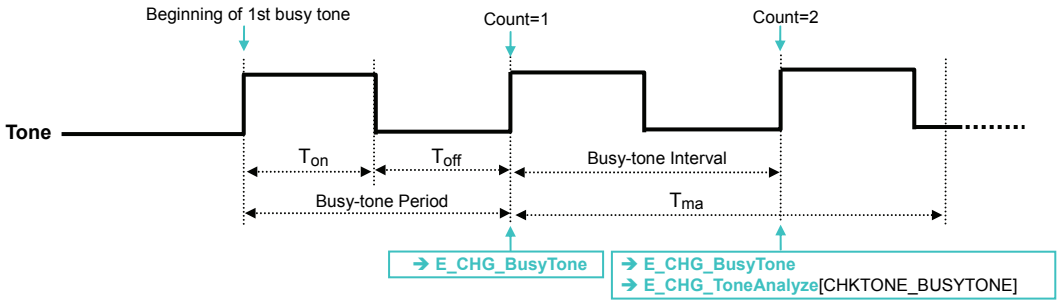
对于第 1 呼叫进程音检测器的回铃音检测器，配置项RingEchoTonePara或函数SsmSetRingEchoTonePara用于设置T_{on}和T_{off}；对于第 2 呼叫进程音检测器的回铃音检测器（如果启用），配置项2ndRingEchoTonePara或函数SsmSet2ndRingEchoTonePara用于设置第 2 呼叫进程音检测器的T_{on}和T_{off}。

每当第 1 呼叫进程音检测器或第 2 呼叫进程音检测器中的任何一个回铃音检测器的回铃音计数器值发生变化时，驱动程序都会抛出E_CHG_RingEchoToneTime事件；而只有当回铃音计数器值变化为 1 时，驱动程序才抛出E_CHG_ToneAnalyze事件(携带CHKTONE_ECHOTONE参数)。函数SsmGetRingEchoToneTime也可用于获取回铃音的保持时间。

1.10.5.3.2.1 忙音检测器

1.10.5.3.2.2 常规功能描述

忙音的检测由忙音检测器完成。典型的忙音波形如下图所示：



一个忙音周期分为高电平（T_{on}）部分和低电平部分（T_{off}），由于忙音信号的占空比为 50%，因此 T_{on} 和 T_{off} 的保持时间相同，驱动程序也只使用忙音周期来表达忙音的特征参数，如上图所示。忙音周期允许的误差为±20%。

当输入信号的高、低电平持续时间与设置的忙音参数相匹配时，忙音检测器就会判定为忙音，相应的忙音计数器（上图中的 Count）就加 1，同时抛出 E_CHG_BusyTone 事件。当忙音计数器的值等于某个指定的值时，忙音检测器会抛出 E_CHG_ToneAnalyze 事件（带 CHKTONE_BUSYTONE 参数）。忙音检测器的结果也可以通过下列函数获取：

函数名称	功能简述
SsmGetBusyToneCount	获取第 1 呼叫进程音检测器输出的忙音周期的个数
SsmGet2ndBusyToneCount	获取第 2 呼叫进程音检测器输出的忙音周期的个数
SsmGetBusyToneLen	获取第 1 呼叫进程音检测器输出的忙音保持时间
SsmGet2ndBusyToneLen	获取第 2 呼叫进程音检测器输出的忙音保持时间

每个忙音检测器可以同时对 4 组具有不同忙音周期的忙音信号进行检测。

第 1 呼叫进程音检测器的忙音检测器的工作参数设置：

- 配置项BusyTonePeriod（或SsmSetBusyTonePeriod、SsmSetBusyTonePeriodEx函数）用来设置忙音周期；
- 配置项IsBusyToneDetermineCount（或SsmSetIsBusyToneDtrCnt函数）可以用来设置驱动程序在检测到第几个忙音时抛出E_CHG_ToneAnalyze事件；

第 2 呼叫进程音检测器的忙音检测器的工作参数设置：

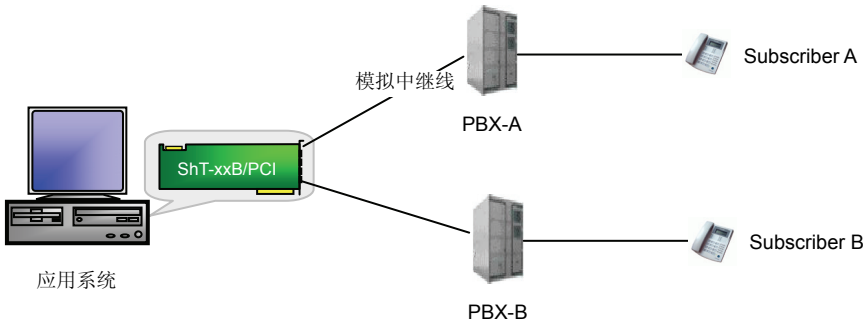
- 配置项2ndBusyTonePeriod（或SsmSet2ndBusyTonePeriod、SsmSetBusyTonePeriodEx函数）可以用来设置忙音周期；
- 配置项2ndIsBusyToneDetermineCount（或SsmSet2ndIsBusyToneDtrCnt函数）可以用来设置驱动程序在检测到第几个忙音时抛出E_CHG_ToneAnalyze事件；

在实际的应用中，由于语音信号中偶尔会出现类似于忙音特征的信号，从而导致驱动程序可能会对忙音进行误判。忙音检测器通过设置一个保护时间门限（记为 T_{max}）来解决这种问题。当两个忙音周期之间的间隔大于 T_{max} 时，忙音计数器自动清零，从而避免了忙音的误检测。

配置项MaxBsTnOffTime用来设置忙音检测的保护时间T_{max}。第 1、第 2 呼叫进程音检测器的忙音检测器均使用此配置项。

1.10.5.3.2.3 背靠背忙音检测功能

对于模拟中继通道，在某些特殊的应用场合下，常规的忙音检测方法可能会失效。例如，对于下图中所示的基于 SHT 系列板卡的应用系统：



用户 A 通过 PBX-A 接入到应用系统的 SHT 系列板卡的中继通道，用户 B 通过 PBX-B 接入到应用系统的 SHT 系列板卡的中继通道，应用系统通过交换总线实现了用户 A 和用户 B 之间的双向通话。当其中一方挂机后（比如用户 A），PBX-A 会向板卡发送忙音；用户 B 听到忙音后也挂机，从而导致 PBX-B 也向板卡发送忙音。这时候如果板卡的电路交换尚未清除，由于回波的影响，任何一个通道上的忙音波形就会发生畸变，不再具有常规忙音的波形特征，忙音检测器就无法正确地检测出忙音，从而导致这 2 个通道被永远占用。

为了解决这个问题，忙音检测器增加了背靠背忙音检测的功能，当发生这类情况时仍然能够正确地检测出忙音。背靠背忙音检测功能无需配置，但其结果输出不是放在信号音检测器的结果输出中，而是单独E_CHG_BusyToneEx事件提供给应用程序。函数SsmGetBusyToneEx也可以用来获得背靠背忙音检测的结果。

注意：只有第 1 呼叫进程音检测器才具有背靠背忙音检测的功能。

1.10.5.3.3 自定义信号音检测器（User-defined Tone Detector）

自定义信号音检测器只用于某些特殊的场合，比如，在拨号音检测器、回铃音检测器和忙音检测器工作时，还需要同时对一些用户自定义的波形进行检测。自定义信号音检测器可以检测连续信号音和周期性信号音。

第 1 呼叫进程音检测器的自定义信号音检测器的工作参数设置：

- 配置项AppointedToneAnalyzerSwitch用于设置工作状态；
- 配置项IsAppointedToneDetermineTime用来设置连续信号音的最小持续时间。判别原理与拨号音检测器相同。当信号音实际持续时间大于此配置项指定门限时，驱动程序输出E_CHG_ToneAnalyze事件（带CHKTONE_APPOINTEDTONE参数）；
- 配置项AppointedTonePara、IsAppointedToneDetermineCount用来设置周期性信号的高电平持续时间和低电平持续时间、最小计数周期。判别原理与回铃音检测器相同。当信号音实际周期计数大于IsAppointedToneDetermineCount指定的最小计数周期时，驱动程序输出E_CHG_ToneAnalyze事件（带CHKTONE_APPOINTEDTONE参数）；

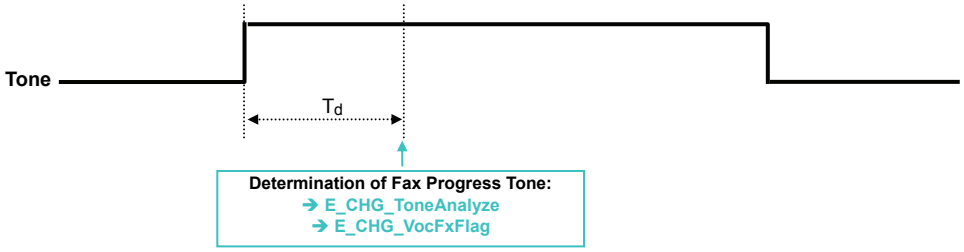
第 2 呼叫进程音检测器的自定义信号音检测器的工作参数设置：

- 配置项2ndAppointedToneAnalyzerSwitch用于设置工作状态；
- 配置项2ndIsAppointedToneDetermineTime用来设置连续信号音的最小持续时间。判别原理与拨号音检测器相同。当信号音实际持续时间大于此配置项指定门限时，驱动程序输出E_CHG_ToneAnalyze事件（带CHKTONE_APPOINTEDTONE参数）；
- 配置项2ndAppointedTonePara、2ndIsAppointedToneDetermineCount用来设置周期性信号的高电平持续时间和低电平持续时间、最小计数周期。判别原理与回铃音检测器相同。当信号音实际周期计数大于2ndIsAppointedToneDetermineCount指定的最小计数周期时，驱动程序输出E_CHG_ToneAnalyze事件（带CHKTONE_APPOINTEDTONE参数）；

1.10.5.4 传真进程音检测器(CTI 系列)

传真进程音检测器共有 2 个，称为第 1 传真进程音检测器和第 2 传真进程音检测器，用于同时检测 2 个具有不同频率特征的传真进程音。传真进程音为单音频、连续性的信号音。

传真进程音的波形特征如下图所示：



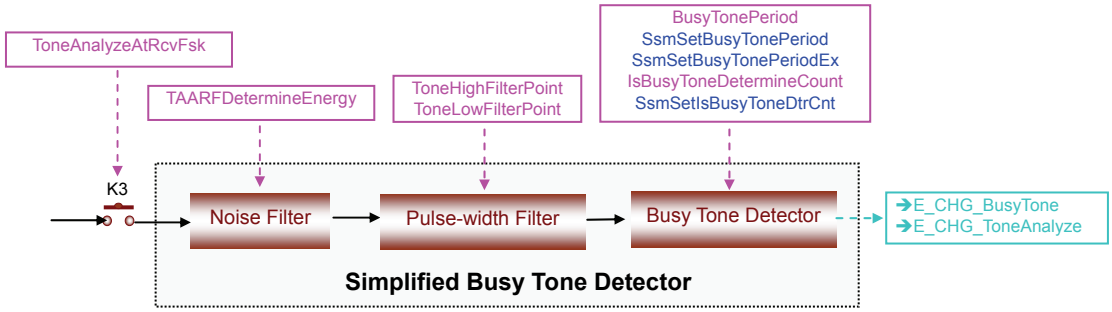
传真进程音检测器在检测到符合频率特征的信号音后开始计时，当计时达到判别门限T_d后，驱动程序就会顺序抛出E_CHG_VocFxFlag、E_CHG_ToneAnalyze事件(带CHKTONE_VOICEF1 或CHKTONE_VOICEF2 参数) 。应用程序也可以通过函数SsmGetVocFxFlag获取检测结果。

传真进程音检测器的工作参数包括中心频率、带宽、带内能量占全频能量的百分比门限、最小保持时间T_d，第 1 传真进程音检测器的工作参数可以通过配置项VoiceFreqF1Para或函数SsmSetVoiceFxPara进行设置，第 2 传真进程音检测器的工作参数可以通过配置项VoiceFreqF2Para或函数SsmSetVoiceFxPara进行设置。

注意：对于模拟中继线类型的通道，在 AutoDial 任务的执行过程中，如果在线路上检测到了传真机的应答信号音，通常表明被叫（传真机）已经摘机，因此传真进程音检测器的检测结果会自动作为状态机的输入事件。

1.10.5.4.1 简易忙音检测器(Simplified Busy Tone Detector)

简易忙音检测器的原理框图、相关配置项或函数、输出事件如下图所示：



简易忙音检测器的工作状态由K3 开关控制，可以通过配置项ToneAnalyzeAtRcvFsk设置。

噪声滤波器的工作原理与前面的“噪声滤波器”节完全一样，只是噪声判别门限是通过配置项TAARFDetermineEnergy来设置。在大多数情况下，噪声滤波器使用缺省配置即可，无须配置。

脉宽滤波器的工作原理与前面章节中“呼叫进程音检测器”中的“脉宽滤波器”部分完全一样，所用的配置项也完全相同。

与呼叫进程音检测器所不同的是，简易忙音检测器不检测输入信号的频率特征，即假定输入信号必定是信号音，因此不包含类似于呼叫进程音检测器中的频率检测器模块。

忙音的波形特征检测算法与呼叫进程音检测器中的常规忙音检测算法相同，并且同样可以同时对 4 组具有不同周期的忙音信号进行检测，波形特征的工作参数也使用第 1 呼叫进程音检测器中的忙音检测器所使用的波形参数，即

- ✧ 配置项BusyTonePeriod或函数SsmSetBusyTonePeriod、SsmSetBusyTonePeriodEx用来设置忙音周期；
- ✧ 配置项IsBusyToneDetermineCount或函数SsmSetIsBusyToneDtrCnt用来设置驱动程序在检测到第几个忙音时抛出E_CHG_ToneAnalyze事件；

简易忙音检测器也具有忙音抗误检测功能，也是通过配置项MaxBsTnOffTime用来设置忙音检测的保护时间 T_{max}。

简易忙音检测器的输出与标准忙音检测器的输出方式完全一样。每当检测出一个忙音周期，忙音计数器（上图中的 Count）就加 1，同时抛出 E_CHG_BusyTone 事件。当忙音计数器的值等于某个指定的值时，忙音检测器会抛出 E_CHG_ToneAnalyze 事件（带 CHKTONE_BUSYTONE 参数）。

注意：

- **如果是模拟中继通道，E_CHG_ToneAnalyze 事件会同时被自动传递给该通道的呼叫状态机进行处理；**
- **只有 SHT 系列的模拟中继通道才有简易忙音检测器。**

1.10.5.5 SIT 检测器(Special Information Tone Detector)

1.10.5.6 AMD (Answering Machine Detector)

在电话呼出过程中，当完成被叫号码的发送，并且被叫方摘机后， AMD 用于检测应答方是电话录音，还是真正的人工应答。

AMD 有 2 种工作模式：

- ✧ VAD 模式：即语音活动检测器模式（Voice Activity Detector），不区分机器应答与人工应答，如果来话信号不是噪声，但又不属于任何一种信号音时，就认为是人工应答；
- ✧ AMD 模式：需要区分机器应答与人工应答。在 AMD 模式下，区分机器应答与人工应答的准确率大约为≥86%。

AMD 的工作模式可以通过配置项进行设置。

AMD 的检测结果通过 E_CHG_ToneAnalyze 事件输出，并且通过携带的参数来区分检测结果：

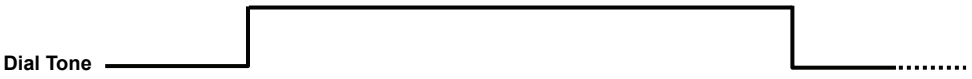
- ✧ 携带 CHKTONE_ANS_MAC 参数： 检测到机器应答。
- ✧ 携带 CHKTONE_VOICE 参数： 检测到人工应答。

1.10.6 信号音发生器(Tone Generator)（CTI 系列）

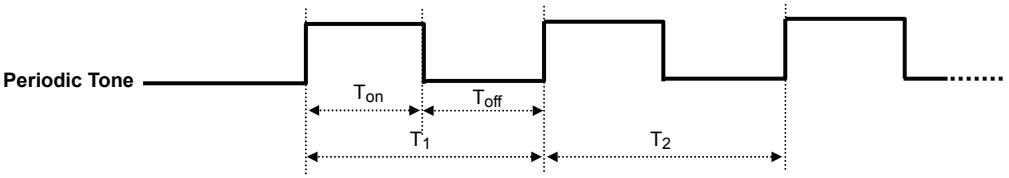
信号音是具有某种恒定频率的、符合一定节拍的呼叫进程音。呼叫进程音采用不同的节拍来表示不同的线路状态，如对端挂机、线路忙等。三汇板卡的每个通道都具有独立的信号音发生器（Tone Generator）。

信号音发生器的工作参数包括中心频率（可以是单音频或双音频）、信号幅度和节拍。根据节拍的变化，信号音可以分为持续性信号音和周期性信号音。

持续性信号音是指信号音发生器一旦启动后，线路上的输出就保持高电平，直到信号音发生器被关闭。拨号音就属于持续性信号音，其波形特征如下图所示：



周期性信号音是指信号音由数个信号周期组成，每个周期由一段持续的高电平和一段持续的低电平组成。忙音和回铃音属于周期性信号音。周期性信号音的典型波形如下图所示：



上图中，T_1、T_2 分别为信号音的第 1 个、第 2 个周期；T_{on} 和 T_{off} 分别为一个信号音周期内的高电平持续时间和低电平持续时间。T_{on} 与 T_{off} 的比值称为占空比。忙音的占空比通常为 1：1，回铃音的占空比通常为 1：4。

1.10.6.1 设置信号音发生器的工作参数

与频率和幅度相关的配置项或函数如下表所示：

函数名称	简要描述	相关配置项
SsmQueryOpSendTone	查询通道是否支持信号音发生器的操作	

SsmSetTxTonePara	设置每个通道上信号音发生器的频率和幅度	DefaultSendToneFrequence DefaultSendToneVolume
SsmGetTxTonePara	获取每个通道上信号音发生器的频率和幅度	

1.10.6.2 产生信号音

与信号音发生器相关的函数如下表所示：

函数名称	简要描述
SsmSendTone	启动信号音发生器
SsmSendToneEx	启动信号音发生器（可在参数中设置 T_{on} 与 T_{off}）
SsmStopSendTone	关闭信号音发生器
SsmChkSendTone	获取信号音发生器正在发送的信号音类型

注意：一旦启动了信号音发生器后，不能执行下列操作：

- 语音播放；
- DTMF 发生器；
- 发送 FSK 数据。

1.10.7 回波抵消器（CTI 系列）

回波抵消功能可以改善通话的质量，大幅提升 DTMF 检测器、Barge in 检测器的准确性。

回波抵消器的工作状态可以通过函数SsmSetEchoCanceller或配置项EnableEchoCancellor进行设置，缺省值为开启。**注意：由于回波抵消功能开启时会改变来话信号，因此，如果需要在通道上进行FSK数据接收操作，请关闭回波抵消器，待FSK数据接收操作完成后再重新开启，以保证数据通信的可靠性。**

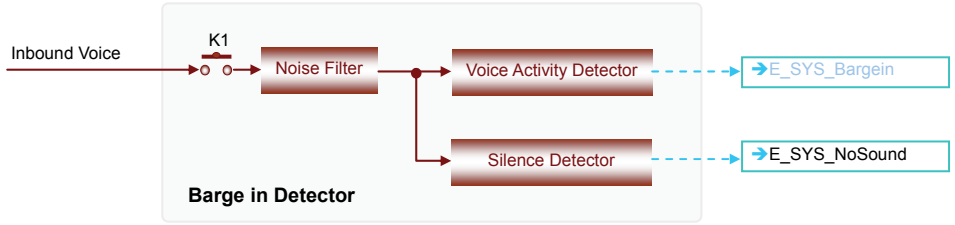
对于SHD系列的 30 通道、60 通道和 120 通道规格的板卡，通过修改配置项LoadFskBin的设置，可以增强回波抵消器的性能和效果，但由于DSP的资源耗尽，因此会丧失录音功能和FSK收/发功能。

Synway板卡提供的回波抵消器的工作模式和参数的缺省值已经能够满足绝大多数应用场景，只有在某些特殊的场合，才需要调整回波抵消器的工作模式和工作参数，请在Synway公司技术人员的指导下进行。回波抵消器的工作参数可以通过函数SsmSetEchoCancelDelaySize或配置项EchoCancelDelaySize进行设置。

1.10.8 Barge in 检测器（Barge in Detector）

Barge in 检测器用于检测线路上的语音活动，通常应用于声控录音、自动语音识别（ASR）等场合。如果通道具有回波抵消能力，由于本端去话信号（例如，放音操作时产生的语音）的回波经回波抵消器后不会进入 Barge in 检测器，Barge in 检测器的检测结果就会非常准确。通道是否具有 Bargein 检测器，请查询相关的硬件说明资料。

Barge in 检测器由噪声滤波器（Noise Filter）、语音活动检测器（Voice Activity Detector）和静默检测器（Silence Detector）组成，如下图所示：



开关 K1 控制 Barge in 检测器的工作状态，由驱动程序自动控制。驱动程序的控制机制为：

通道类型	驱动程序的动作	触发条件
需要参与会议调度的会议通道	闭合 K1	通道加入会议室

	断开 K1	退出会议室
TUP / ISUP 通道 No.1 信令通道	闭合 K1	通道状态迁移到 S_CALL_TALKING
	断开 K1	通道状态迁移出 S_CALL_TALKING
ISDN 通道	闭合 K1	通道状态迁移到 S_CALL_WAIT_REMOTE_PICKUP（去话呼叫）或 S_CALL_RINGING（来话呼叫）
	断开 K1	通道状态迁移出 S_CALL_TALKING
模拟中继线通道	闭合 K1	通道状态迁移出 S_CALL_STANDBY
	断开 K1	通道状态迁移到 S_CALL_STANDBY
坐席通道	闭合 K1	通道状态迁移到 S_CALL_PICKUPED
	断开 K1	通道状态迁移到 S_CALL_STANDBY
模拟中继线录音通道	闭合 K1	检测到摘机
	断开 K1	检测到挂机

对于SHD系列板卡，如果希望通道处于任何状态时，K1 开关都保持闭合状态，可以通过配置项AlwaysDetectBargeIn实现。

对于SHN系列板卡，开关K1 的控制仅由配置项AlwaysDetectBargeIn实现，驱动程序不会进行自动控制。

噪声滤波器用于区分线路噪声和真实的语音信号。配置项VoiceEnergyMinValue用来设置噪声滤波器的噪声门限，当来话信号或去话信号的强度低于设定值时，会被判定为线路噪声。函数SsmSetBargeInSens或配置项BargeInSensitive用来设置噪声滤波器的灵敏度。当来话中包含了DTMF信号时，是否将DTMF信号视为语音信号，取决于配置项DefaultDtmfIsSound的设定值。

语音活动检测器用于检测线路上的语音活动。当来话信号中检测到了语音活动，并且保持时间超过函数SsmGetIsBargeInDtrmTime 或配置项 BargeInDtrmTime 的设定值时，输出E_SYS_BargeIn 事件。函数SsmDetectBargeIn也可以用来获取检测结果。

静默检测器用于判断线路上是否保持静默。当来话信号保持静默的时间超过函数SsmSetNoSoundDtrmTime或配置项IsNoSoundDtrmTime的设定值时，输出E_SYS_NoSound事件。函数SsmDetectNoSound可以用来获取静默的检测结果，函数SsmGetNoSoundTime用来获取静默的保持时间。

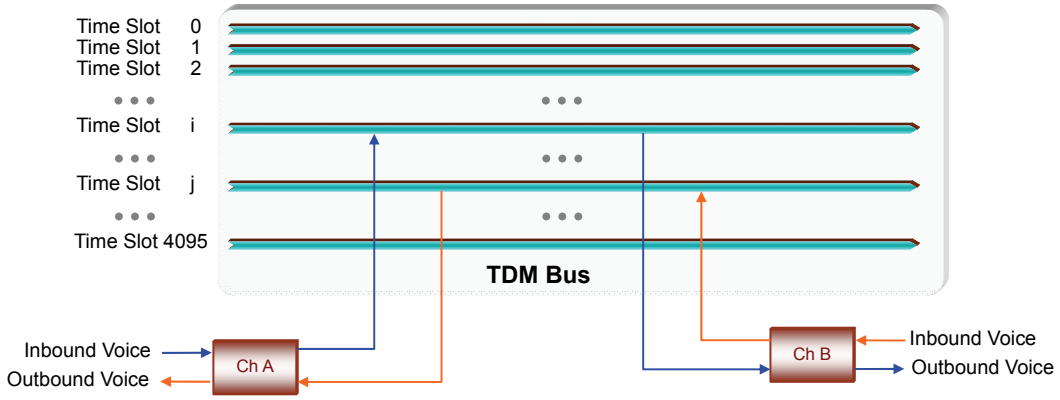
1.10.9 TDM 交换总线

Synway 板卡支持 2 种模式的语音交换能力：

- 内部交换能力：板卡内部的通道之间的语音交换能力。所有的 Synway 板卡都具有内部交换能力
- 外部交换能力：分布在不同板卡上的通道之间的交换能力。除 4 通道（-USB 型号）和 8 通道规格的 SHT、ATP 系列板卡外，所有 Synway 板卡都具有外部交换能力。

cPCI 总线规格的板卡采用 H.110 总线，其余板卡采用 H.100 总线。

使用 TDM 交换总线进行语音交换的原理如下图所示：



TDM 交换总线的原理

上图中，建立通道 A 与通道 B 的双向连接的操作过程为：Ch A 的来话上行到 TDM 总线的时隙 i(Time Slot i)，将时隙 i 下行到 Ch B，就实现了将 Ch A 的来话送给 Ch B 听的功能；同样，Ch B 的来话上行到 TDM 总线的时隙 j(Time Slot j)，将时隙 j 下行到 Ch A，就实现了将 Ch B 的来话送给 Ch A 听的功能。

系统初始化时，如果配置项InVoiceToBus设置为 1（缺省值），驱动程序会将通道的来话信号连接到TDM总线的某个时隙上。

常用的 TDM 总线交换的函数包括：

交换类别	函数名称	功能简述
建立双向连接	SsmTalkWith	建立 2 个通道之间的双向连接
	SsmTalkWithEx	建立 2 个通道之间的双向连接，可以设置参与交换的 2 个通道的音量
拆除双向连接	SsmStopTalkWith	拆除 ch1 到 ch2 的双向连接
建立单向连接	SsmListenTo	建立说者通道到听者通道的单向连接
	SsmListenToEx	建立说者通道到听者通道的单向连接，可设置说者通道的音量
	SsmLinkFrom	建立说者通道到听者的单向连接，允许听者通道同时听多个通道
	SsmLinkFromEx	建立说者通道到听者通道的单向连接，允许听者通道同时听多个通道，并且可设置说者通道的音量
	SsmLinkFromAllCh	批量建立一个说者通道到多个听者通道的单向连接，允许听者通道同时听多个通道，并且可设置说者通道的音量
拆除单向连接	SsmStopListenTo	拆除用函数 SsmListenTo 或 SsmListenToEx 建立的单向连接
	SsmStopLinkFrom	拆除用函数 SsmLinkFrom 或 SsmLinkFromEx 建立的单向连接
	SsmUnLinkFromAllCh	拆除用函数 SsmLinkFromAllCh 建立的单向连接
拆除全部连接	SsmClearChBusLink	拆除通道上建立的所有总线连接，使通道恢复到驱动程序初始化完成时的总线连接状态

高级的 TDM 总线交换函数用于建立或拆除通道与总线时隙之间的单向连接，包括下列函数：

交换类别	函数名称	功能简述
通道上行到总线时隙	SsmLinkToBus	改变上总线混音器的输出信号上行到总线的时隙
	SsmUnLinkToBus	恢复将上总线混音器的输出信号连接到缺省的总线时隙
总线时隙下行到通道	SsmLinkFromBus	建立总线时隙到听者通道的单向连接
	SsmLinkFromBusEx	建立总线时隙到听者通道的单向连接，并可设置音量
	SsmUnLinkFromBus	拆除总线时隙到听者通道的单向连接

1.10.10 分布式电话会议(CTI 系列)

1.10.10.1 基本概念

- **分布式电话会议技术**

CTI 系列板卡上的每个语音通道都具有独立的混音器，不需要借助额外的电话会议板卡，就可以实现电话会议功能，称为分布式电话会议。分布式电话会议技术是 Synway 公司的专利技术。

- **会议室**

会议室由参与同一个会议的通道组成。每个会议室具有一个唯一的编号，可由应用程序指定，也可由驱动程序执行分配。一个会议室中通道的总数最少可以为 3 个，最多可以达到整个应用系统的通道总数，可以通过配置项ConfDefaultMaxGroupMember进行设置，或者在调用函数SsmCreateConfGroup创建会议室时在参数中指定。

会议室的个数可以通过配置项ConfMaxGroup进行设置。

- **会议室编号**

每个会议室都有一个唯一的编号，称为会议室编号。会议室编号从 0 开始，最大编号为配置项ConfMaxGroup的设定值减 1。

- **会议通道**

加入某个会议的通道称为会议通道。

- **会议通道的编号**

会议通道在会议室中的逻辑编号，从 0 开始编号。注意：会议通道的编号与通道编号不一定相同。

- **会议调度程序**

由于会议混音器的输入信号源的数量是有限的，因此，对于成员数量较多的会议室，当同时发言的通道数量超过会议混音器允许的最大值时，必然有一部分通道的声音无法进入会议混音器。SynCTI 驱动程序包含了一个会议调度程序，可以对会议室中的发言情况进行自动调度。只有满足调度条件的通道，会议调度程序才会允许其声音信号进入会议混音器。

会议调度算法以会议通道的调度优先级作为调度依据。在分配会议混音器资源时，首先满足优先级高的通道，最后满足优先级低的通道，直到全部会议混音器资源耗尽为止。

假设可供调度的会议混音器资源为 N，主持人通道数量为 m，主席通道数量为 n，动态发言通道中正在发言的通道数量为 k，会议调度算法的调度规则为：

(1) 首先为 m 个主持人分配会议混音器。分配完毕后，如果还有剩余资源，进入下一步。

(2) 如果 n≤N-m，所有主席通道均可以获得会议混音器；如果 n>N-m，调度程序将全部 n 个主席通道的语音信号强度（即嗓门大小）按照从高到低的方式进行排序，前面 N-m 个将获得会议混音器。如果处理完主席通道后还有多余的混音器资源，进入下一个优先级的分配。

(3) 如果 k≤N-m-n，所有正在发言的 k 个动态发言通道均获得会议混音器；如果 k> N-m-n，调度程序将全部 k 个动态发言通道的语音信号强度按照从高到低的方式进行排序，排在前面的 N-m-n 个会议通道将获得会议混音器。

(4) 任何一个会议通道获得了会议混音器的使用权后，如果没有更高优先级的通道需要使用会议混音器，它将保持对会议混音器的占用状态，直到它重新回到静默状态并保持一定的时间后，才会重新参与调度。静默保持时间可以通过配置项 ConfDefaultMaxSilenceTime 进行设定，也可以在调用函数 SsmCreateConfGroup创建会议室时在参数中指定。

- **会议通道的发言模式**

会议通道加入到会议室时，需要给通道分配发言权限。SynCTI 驱动程序支持 5 种发言模式，如下表所示：

发言模式	描述
主持人	可以发言，也可以听取其它通道的发言。此模式具有最高的调度优先级。 具有主持人发言模式的会议通道简称为主持人通道。建议应用程序为每个会议室只设置 1

	个主持人通道。
主席	可以发言，也可以听取其它通道的发言。此模式的调度优先级低于主持人模式，但高于动态发言模式。 具有主席发言模式的会议通道简称为主席通道。
动态发言	可以发言，也可以听取其它通道的发言。此模式的调度优先级低于主席模式。 具有动态发言模式的会议通道简称为动态发言通道。
听众	不能发言，只能听取其它通道的发言。听众通道不参与会议调度。 具有听众发言模式的会议通道简称为听众通道。
背景音乐	只能发言，但不能听取其它通道的发言，一般用于向会议室播放背景音乐。**注意：此模式的通道是否参与会议调度、以何种优先级进行调度，取决于配置项BackgroundVoicePriority的设置。** 具有背景音乐发言模式的会议通道简称为背景音乐通道。
动态只发言	可以发言，但不能听取其它通道的发言。 具有动态发言模式但不能听取其他通道发言的会议通道简称为动态只发言通道。

- **DTMF 钳位（DTMF Clamping）**

当参与会议的人在话机上进行按键操作时，线路上的 DTMF 信号音会进入会议混音器，被其它会议通道听见。Synway 驱动程序支持 DTMF 钳位功能，可以有效过滤按键产生的 DTMF 信号。

DTMF 钳位功能的实现原理是：当板卡上的 DTMF 检测器在会议通道上检测到 DTMF 信号时，会立即断开该通道的来话信号；当 DTMF 信号消失后，自动恢复来话信号。

配置项ClearInVoiceOnRxDtmf或函数SsmSetFlag（携带F_ClearInVoiceOnRcvDtmf参数）用于设置是否开启DTMF钳位功能。

- **禁止发言功能(Coming Voice Blocking)**

会议通道加入会议室后，在正常情况下，只有会议通道满足调度规则，它的来话信号才可以进入会议混音器，被其它会议通道听见。如果需要临时禁止该通道的发言，可通过配置项InVoiceToBus或函数SsmSetFlag（携带F_InVoiceToBus参数）实现。

1.10.10.2 管理会议室

下表是有关会议室的函数和配置项的一览表：

函数名称	类别	功能描述
SsmCreateConfGroup	函数	创建一个会议室
SsmFreeConfGroup	函数	注销一个会议室
ConfDefaultMaxSilenceTime ConfDefaultMaxGroupMember ConfDefaultMaxGroupSpeaker ConfDefaultMaxGroupSpeaking	配置项	设置会议室的工作参数
SsmGetConfCfgInfo SsmGetTotalConfGroup SsmGetConfGrpInfo SsmGetConfGrpCfgInfo SsmGetConfGrpId SsmValidateGrpId	函数	获取会议室的相关信息
SsmGetConfGrpMmbrId SsmGetConfGrpMmbrInfo SsmGetConfChInfo	函数	获取会议室中成员的相关信息

1.10.10.3 管理会议通道

有关会议通道的相关函数和配置项如下表所示：

函数名称	类别	功能描述
SsmJoinConfGroup	函数	将通道加入到会议室
SsmExitConfGroup	函数	将通道退出会议室
SsmSetListenVlmInConf	函数	设置通道听到的会议声音的音量

1.10.10.4 给会议室播放背景音乐

如果需要给会议室播放背景音乐、会议通知等信息，实现方式为：

- 选择一个执行语音播放任务的会议通道。选择通道的方法包括：
  - 为会议室单独设置一个背景音乐通道。这种方式非常灵活，可以按照实际需要设置通道的发言优先级，但缺点是需要占用一个额外的通道；
  - 在会议室中选择一个主持人通道。如果会议室中有主持人发言模式的会议通道，这是最合理的解决方案，但在编写程序时应尽量让该通道最后退出会议室。
  - 如果会议室中没有主持人会议通道，可选择一个主席通道或动态发言通道。建议优先选择主席通道。这种方式的缺点是，如果执行播放任务的会议通道因某种原因（如连接意外断开等）中途退出会议室时，应用程序需要将播放任务转移到其它会议通道上。
- 调用函数SsmSetPlayDest闭合SHT/SHD/SHN系列板卡原理框图中的k1-1 开关，更多信息请参见本章中“SHT系列板卡的工作原理框图”或“SHD系列板卡的工作原理框图”或“SHN系列板卡的工作原理框图”部分内容；
- 调用语音播放的函数，更多信息请参见本章中“语音播放”部分内容。

1.10.10.5 录制会议室的声音

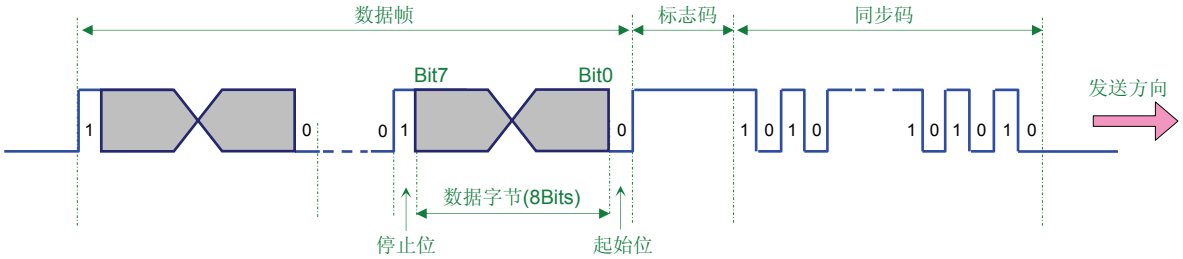
如果需要录制会议室播的声音，实现方式为：

- 选择一个执行录音任务的会议通道。任何一个会议通道都可以用来执行录音任务，但应尽可能选择中途不会退出会议室的通道，以避免在通道之间转移录音任务。建议的通道选择方法包括：
  - 优先选择用于播放背景音乐的会议通道；
  - 选择一个主持人通道；
  - 选择一个主席通道；
  - 选择一个动态发言通道；
  - 选择一个听众通道或动态发言通道。
- 如果执行录音任务的会议通道同时也执行播放背景音乐的任务，并且是SHD/SHN系列板卡上的通道，则应调用函数SsmSetRecBack闭合“SHD系列板卡的工作原理框图”或“SHN系列板卡的工作原理框图”中的k6-1 开关和k6-2 开关；
- 调用语音录制的函数，更多信息请参见本章中“录音操作”部分内容。

1.11 FSK 数据收发器（CTI 系列）

BFSK（Frequency Shift Keying），是采用不同的载波频率表示二进制数据的 0 和 1 的数字通信技术，通常用于在电话线上传送主叫号码、短信等信息。

BFSK 的数据流以帧为单位，帧格式如下图所示：



一段完整的 BFSK 数据流由同步码、标志码和数据三部分组成。同步码是 0/1 翻转的波形，标志码为高电平 1。每个数据字节的前面会插入起始位 0，后面会插入停止位 1。每个数据字节的比特 0 最先发送，比特 7 最后发送。

Synway 的 CTI 系列语音卡为每个语音信道均配备了 FSK 数据收发器，采用半双工工作模式，波特率为 1200bps，比特 1 对应的频率是 1200Hz，比特 0 对应的频率是 2200Hz。

1.11.1 FSK 数据发送器

FSK 数据发送器的相关函数、配置项和事件如下表所示：

类别	名称	描述
配置项	FreqBit0 FreqBit1 Baudrate MdlAmp	设置 FSK 数据发送器工作参数
函数	SsmSetFskPara	设置 FSK 数据发送器的工作参数
	SsmTransFskData	将数据转换为 BFSK 格式的二进制数据流
	SsmStartSendFSK	启动 FSK 数据发送器
	SsmCheckSendFsk	查询 FSK 数据发送器的进展情况
	SsmStopSendFsk	停止 FSK 数据发送器
事件	E_PROC_SendFSK	当驱动程序完成全部数据的发送后，会向应用程序抛出此事件

1.11.2 FSK 数据接收器

Synway 板卡的 FSK 数据接收器的波特率 1200bps，比特 0 的载波频率为 2200Hz，比特 1 的载波频率为 1200Hz，均不可通过函数或配置项进行修改。

有关 Fsk 接收器的函数、事件和配置项如下表所示：

名称	描述
配置项：ToneAnalyzeAtRcvFsk	设置 Fsk 数据接收器工作时是否分析信号音
配置项：FskMarkSignal	设置 Fsk 数据接收器判定接收数据的条件
配置项：FskEchoCancelDelay 函数：SsmSetFlag（携带 F_EchoCancelInFsk参数）	当 FSK 接收器处于工作状态时，是否关闭回波抵消器
函数：SsmStartRcvFSK 函数：SsmStartRcvFSK_II 函数：SsmStopRcvFSK	启动，停止 Fsk 数据接收器
函数：SsmGetRcvFSK 函数：SsmCheckRcvFSK	获取 Fsk 数据及 Fsk 数据接收器状态查询

函数：SsmClearRcvFSKBuf	清空驱动程序 Fsk 数据接收器缓冲区
事件：E_PROC_RcvFSK	启动 FSK 数据接收器后，驱动检测到接收 FSK 任务结束后，会向应用程序抛出此事件

1.12 传真（CTI 系列）

1.12.1 设置传真资源通道的数量

对于 SHT 系列、SHD 系列板卡，下列型号的板卡支持传真资源通道：

- 型号中带"/FAX"标记的板卡，包括：
  - SHT-8B/PCI/FAX
  - SHT-8C/PCI/FAX
  - SHT-16B-CT/PCI/FAX
  - SHT-16B-CT/cPCI/FAX
  - SHT-16C-CT/PCI/FAX
  - SHD-30B-CT/PCI/SS7/FAX
  - SHD-60B-CT/PCI/SS7/FAX
  - SHD-30B-CT/cPCI/SS7/FAX
  - SHD-60B-CT/cPCI/SS7/FAX
- 型号中不带"/FAX"标记，包括：
  - SHT-16B-CT/PCI
  - SHT-16B-CT/cPCI

由于传真功能和语音功能都是通过DSP芯片上的算法来实现，因此，对于支持传真资源通道的板卡，通过放弃部分或全部通道的语音功能，可以得到更多的传真资源通道。传真资源通道的总数由配置项MaxFaxChannel进行指定。MaxFaxChannel的设定值与板卡型号、可用的传真资源通道总数之间的关系如下表所示：

Board Model	配置项 MaxFaxChannel 的设定值													
	0		4		8		12		16		24		32	
	N_{Fax}	N_{Voc}	N_{Fax}	N_{Voc}	N_{Fax}	N_{Voc}	N_{Fax}	N_{Voc}	N_{Fax}	N_{Voc}	N_{Fax}	N_{Voc}	N_{Fax}	N_{Voc}
SHT-8B/PCI/FAX	0	8	4	8	-	-	-	-	-	-	-	-	-	-
SHT-8C/PCI/FAX	0	8	4	8										
SHT-16B-CT/PCI/FAX	0	16	4	16	8	8	12	0	-	-	-	-	-	-
SHT-16B-CT/PCI	0	16	4	8	8	0	-	-	-	-	-	-	-	-
SHT-16B-CT/cPCI/FAX	0	16	4	16	8	8	12	0	-	-	-	-	-	-
SHT-16B-CT/cPCI	0	16	4	8	8	0	-	-	-	-	-	-	-	-
SHT-16C-CT/PCI/FAX	0	16	4	16	8	8	12	0	-	-	-	-	-	-
SHD-30B-CT/PCI/SS7/FAX	0	30	-	-	-	-	-	-	-	-	24	30	32	0
SHD-60B-CT/PCI/SS7/FAX	0	60	-	-	-	-	-	-	16	60	24	30	32	0
SHD-30B-CT/cPCI/SS7/FAX	0	30	-	-	-	-	-	-	-	-	24	30	32	0
SHD-60B-CT/cPCI/SS7/FAX	0	60	-	-	-	-	-	-	16	60	24	30	32	0

注：上表中，N_{Fax}：可用的传真资源通道总数；N_{Voc}：可用的语音通道总数；-：表示不支持。

1.12.2 传真操作支持的速率

Synway 板卡支持传真操作的型号，以及支持的传真速率如下表所示：

板卡型号	4800bps		9600bps		12000bps		14400bps	
	发送	接收	发送	接收	发送	接收	发送	接收
SHT-8B/PCI/FAX	√	√	√	√			√	
SHT-8C/PCI/FAX	√	√	√	√	√	√	√	√
SHT-16B-CT/PCI/FAX SHT-16B-CT/PCI	√	√	√	√			√	

SHT-16B-CT/cPCI/FAX SHT-16B-CT/cPCI	√	√	√	√			√	
SHT-16C-CT/PCI/FAX	√	√	√	√	√	√	√	√
SHD-30B-CT/PCI/SS7/FAX	√	√	√	√			√	
SHD-60B-CT/PCI/SS7/FAX	√	√	√	√			√	
SHD-30B-CT/cPCI/SS7/FAX	√	√	√	√			√	
SHD-60B-CT/cPCI/SS7/FAX	√	√	√	√			√	

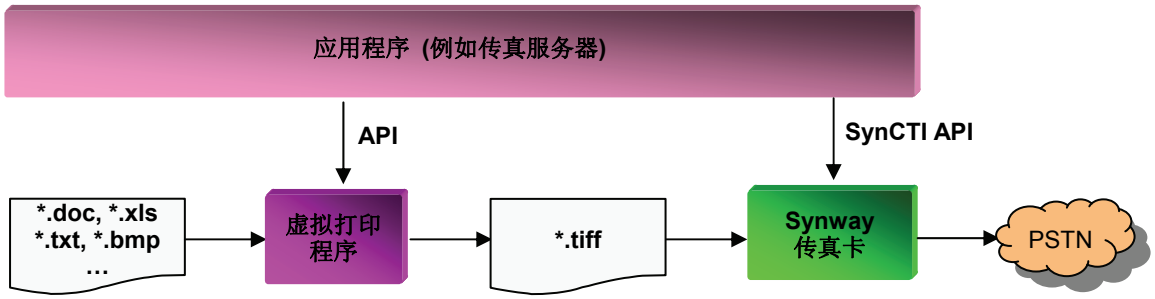
1.12.3 传真操作支持的文件格式

Synway 板卡的传真支持 T.30 通信协议和 T.4 传真图像协议，发送的传真数据采用 MH 编码格式。发送的传真文件支持标准 tiff 文件。

Tiff 文件是用于传真数据保存的主流文件，通用性强，其查看、归档、修改等都比较方便。Synway 传真支持的 Tiff 文件的属性包括：

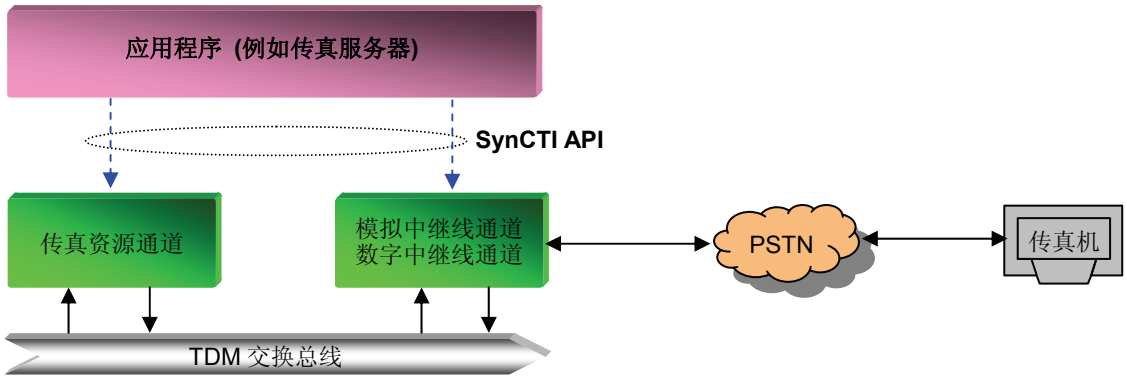
- ✓ 黑白色
- ✓ MH 编码格式
- ✓ 204*196、204 * 98、200 * 200、200 * 100 的分辨率
- ✓ 页面数据是每行 1728 个像素

应用程序发送任意格式文件的工作原理如下图所示：



上图中，虚拟打印程序用于将任何格式的文件转换 tiff 格式。

一次完整的传真发送或接收过程需要使用模拟或数字中继线通道和传真资源通道，如下图所示：



传真的发送（或接收）过程为：

（1） 在模拟或数字中继线通道上发起并建立呼叫
（2） 建立传真资源通道与模拟或数字中继线通道的双向连接
（3） 发送（或接收）传真文件

（4） 拆除呼叫

1.12.4 设置传真操作的工作参数

设置传真操作的工作参数的相关函数如下表所示：

函数名称	功能概述
SsmFaxSetMaxSpeed	传真收/发的初始速率，亦为接收或发送传真时使用的最大速率。此速率对全部通道都起作用。
SsmFaxSetID	设置本方的识别代码。通常为本方的电话号码，也可以设置为字母。这个代码将在接收或发送传真时被传送给对方传真机。

1.12.5 获取传真操作的信息

传真过程中的相关参数可以通过调用以下函数获得：

函数名称	功能概述及获取的信息
SsmFaxGetSpeed	获取传真实际收/发速率。可能值为 24、48、72、96、120、144，分别表示 2400、4800、7200、9600、12000、14400bps。
SsmFaxGetID	获取对方的识别码。一般要在传真握手完成后才能获得。
SsmFaxGetPages	获取传真收发过程中已经完成的页数。如果收/发传真结束后，调用本函数返回零，则表示该次传真收/发过程失败。
SsmFaxGetAllBytes	获取当前传真发送页的总字节数。
SsmFaxGetSendBytes	获取传真发送过程中当前页已经发送的字节数。当调用SsmFaxStartSend函数后，可以用本函数查看当前页已发送的字节数。
SsmFaxGetRcvBytes	获取传真接收过程中当前页已接收的字节数。当调用SsmFaxStartReceive函数后，可以用本函数查看当前页收到的字节数。
SsmFaxGetChStateMsg	获取指定通道上正在进行的传真收/发过程所处的状态信息。
SsmFaxCheckEnd	检查指定通道上的传真接收或发送任务的执行情况。

1.12.6 发送传真

有关传真发送的函数如下表所示：

函数名称	功能概述
SsmFaxStartSend	启动发送传真文件的任务
SsmFaxStartSendEx	启动发送传真文件的任务，可指定起始页和结束页
SsmFaxSendMultiFile	启动发送多个传真文件的任务
SsmFaxSendMultiFileEx	启动发送多个传真文件的任务，可指定起始页和结束页
SsmFaxAppendSend	追加单文件的传真发送，传真资源通道必须处于传真发送状态
SsmFaxCheckEnd	获取传真资源通道的工作状态
SsmFaxSetMaxSpeed	设置传真收发的速率
SsmFaxSetID	设置本方的传真识别码，对方传真的屏幕会显示该识别码
SsmFaxGetID	获取对方的传真识别码
SsmFaxGetAllBytes	获取当前传真发送页的总字节数
SsmFaxGetSendBytes	获取传真发送过程中当前页已经发送出去的字节数
SsmFaxGetSpeed	获取当前传真收发的速度

SsmFaxGetPages	获取传真收发过程中已经完成的页数
SsmFaxStop	强制停止当前的传真过程

1.12.7 接收传真

有关接收传真的函数如下表所示：

函数名称	功能概述
SsmFaxStartReceive	开始传真接收，接收到的传真数据将保存在设定的传真文件中
SsmFaxCheckEnd	获取传真资源通道的工作状态
SsmFaxSetMaxSpeed	设置传真收发的速率
SsmFaxSetID	设置本方的传真识别码，对方传真的屏幕会显示该识别码
SsmFaxGetID	获取对方的传真识别码
SsmFaxGetRcvBytes	获取传真接收过程中当前页已接收的字节数
SsmFaxGetSpeed	获取当前传真收发的速度
SsmFaxGetPages	获取传真收发过程中已经完成的页数
SsmFaxStop	强制停止当前的传真过程

1.13 板载音频功率放大器

ATP 系列、DST 系列、SHT 系列板卡上的 0 通道上都安装了模拟音频放大电路和扬声器输出接口，可以直接连接外置扬声器，将声音送到扬声器进行播放。有关板载音频功率放大器的详细内容请参见相应板卡的原理框图部分。

1.14 编写基于 Synway 板卡的应用程序

1.14.1 设置驱动程序的事件输出模式

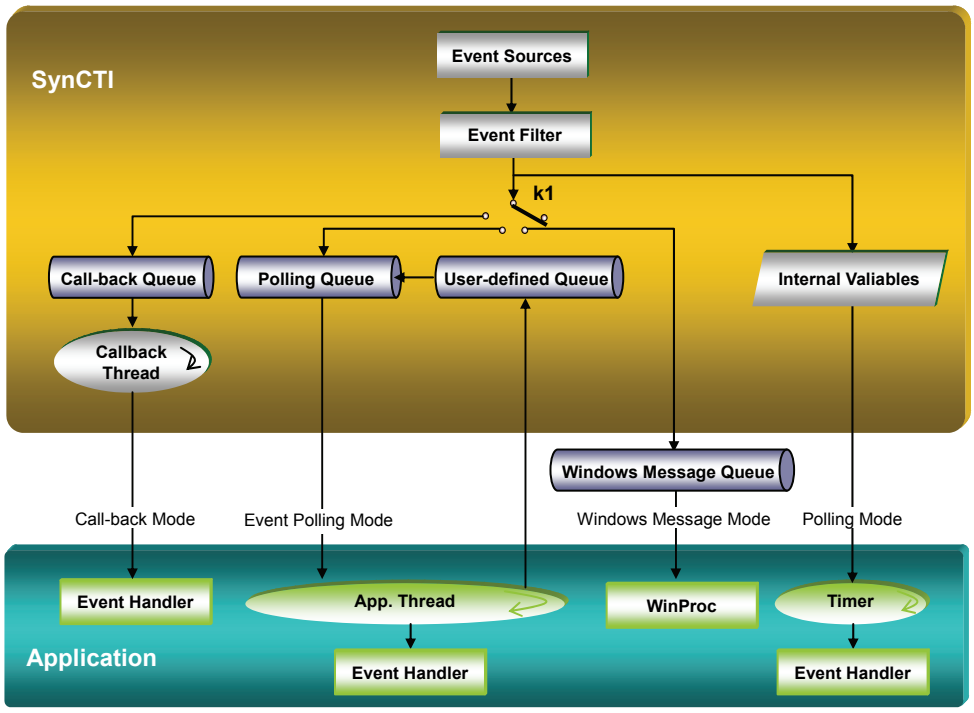
对 Synway 板卡进行编程，一般包括三部分：

- 初始化代码，包括初始化板卡和驱动程序，只需要在应用程序启动时调用一次；
- 业务处理代码，向驱动程序发布命令，对驱动程序通过事件返回的信息进行及时处理，以实现具体的业务；
- 卸载代码，包括释放板卡和驱动程序占用的资源，只需要在应用程序退出时调用一次。

SynCTI 驱动程序支持下列编程模式：

- ✧ 轮询模式：由应用程序不停地调用驱动程序提供的相关查询函数，以获取任务的进展情况。这种编程方式由于消耗计算机的资源较多，效率很低，只能适用于容量较小的应用系统，目前已经被逐渐淘汰。因此，本书不对此编程方式进行更多的说明；
- ✧ 事件等待：应用程序调用驱动程序提供的事件等待函数，当驱动程序没有事件时，应用程序的调用者线程被阻塞；当驱动程序抛出某个事件时，应用程序的调用者线程被重新激活，恢复对事件的处理；
- ✧ 事件回调：由应用程序向驱动程序注册一个回调函数，当驱动程序有事件发生时，由驱动程序调用回调函数，对事件进行处理；
- ✧ Windwos 消息模式：驱动程序将事件发送到 Windows 的消息队列中，通过 Windows 统一的消息队列处理机制，来对事件进行处理。由于 Windows 消息模式可以携带的参数有限，因此实际使用这种编程模式的应用程序并不多，本书也不对此编程模式进行更多的说明。

这四种编程模式的消息过程如下图所示：



开关k1 由函数SsmSetEvent进行控制。SynCTI驱动程序启动后，会自动工作在轮询模式。如果应用程序要使用事件等待方式或事件回调方式，需要调用函数SsmSetEvent来改变驱动程序的工作模式，详细内容请参见后面的编程示例部分内容。

1.14.1.1 事件等待方式的编程示例

```
#include <windows.h>                    // 包含需要的 Windows 头文件
#include "shpa3api.h"                   // 包含 SynCTI 驱动程序需要的头文件
……

void main()
{
    //初始化 SynCTI 驱动程序和板卡
    if ( ! SsmStartCti (……) )
    {                                   //失败
        ……
        MessageBox( "Initialize Voice card failure" );
        return;
    }

    //获取板卡、通道的相关信息
    int nMaxCh = SsmGetMaxCh();                 //获取总的通道数量
    for (int ch=0; ch < nMaxCh; ch ++)
    {
        int nChType = SsmGetChType (ch);        //获取通道类型
        ……
    }

    //设置驱动程序抛出事件的模式
    EVENT_SET_INFO EventMode;
    EventMode.DwWorkMode = EVENT_POLLING;       //使用事件等待方式
    SsmSetEvent(0xffff, 0, 0, &EventMode);

    //事务处理
    MESSAGE_INFO Event;
    While(1)
    {
```

```
        memset(&MessageInfo, 0, sizeof(PMESSAGE_INFO));
        If ( 应用程序退出 ) break;                          //应用程序退出代码

        // 等待并响应驱动程序抛出的事件
        If ( SsmWaitForEvent (50, &Event) == 0 )            //等待事件发生
        {
            switch(Event.EventCode)
            {
            case E_CHG_ChState：                            //通道的状态发生变化
                ……                                          //业务处理代码
                break;
                ……
            case E_XXXXXX：
                ……
                break;
            default:
                break;
            }
        }
    }

    SsmCloseCti();                                          //退出时，关闭驱动程序
}
```

如果需要使用扩展的输出事件的数据结构，可以按照下列方式编写上述代码中的“事务处理”部分：

```
    //事务处理
    UCHAR pucBuffer[300];
    SSM_EVENT SsmEvent;
    while(1)
    {
        memset(&SsmEvent, 0, sizeof(SSM_EVENT));
        memset(pucBuffer, 0, sizeof(UCHAR)* 300);
        SsmEvent.pvBuffer = (PVOID)pucBuffer;
        SsmEvent.dwBufferLength = 300;
        if(SsmWaitForEventA(50, &SsmEvent) == 0)
        {
            switch(SsmEvent.wEventCode)
            {
            case E_CHG_ChState:
                ……                                  //do something
                break;
            case E_RCV_DSTDChannel:
                switch(SsmEvent.dwParam)
                {
                case DST_AUDIO_CHG:
                    ……                              //do something
                    break;
                case DST_MSG_CHG:
                    ……                              //do something
                    break;
                default:
                    break;
                }
            default:
                break;
            }
        }
    }
```

1.14.1.2 事件回调方式的编程示例

```
#include <windows.h>          // 包含需要的 Windows 头文件
#include "shpa3api.h"         // 包含 SynCTI 驱动程序需要的头文件
BOOL bExit = FALSE;           // 控制应用程序退出的变量
……
```

```
//应用程序提供的回调函数，用于处理驱动程序抛出的事件
int CALLBACK MyCallback (WORD wEvent, int nReference, DWORD dwParam, DWORD dwUser)
```

```
{
    switch(wEvent)
    {
    case E_CHG_ChState:
        ……                          //do something
        break;

    case E_XXXXXX：
        ……                          //do something
        bExit = TRUE;               //退出应用程序
        break;

    ……

    default:
        break;


    }
    return 1;
}

//应用程序的主线程
void main()
{
    ……
    if ( ! SsmStartCti (……) )                    //初始化 SynCTI 驱动程序和板卡
    {                                            //失败
        ……
        MessageBox( "Initialize Voice card failure" );
        return;
    }

    int nMaxCh = SsmGetMaxCh();                  //获取总的通道数量
    for (int ch=0; ch < nMaxCh; ch ++)
    {
        int nChType = SsmGetChType (ch);         //获取通道类型
        ……
    }

    //设置驱动程序抛出事件的模式
    EVENT_SET_INFO EventMode;
    EventMode.DwWorkMode = EVENT_CALLBACK;       //事件回调模式
    EventMode.lpHandlerParam = MyCallback;       //注册回调函数
    SsmSetEvent(0xffff, 0, 0, &EventMode);

    While(!bExit) ;                              //等待程序结束

    SsmCloseCti();                               //退出时，关闭驱动程序
}
```

如果需要使用扩展的输出事件的数据结构，可以按照下列方式编写应用程序提供的回调函数：

```
int CALLBACK MyCallback (PSSM_EVENT pEvent)
{
    switch(pEvent->wEventCode)
    {
    case E_CHG_ChState:
        ……                          //do something
        break;

    case E_XXXXXX：
        ……                          //do something
        bExit = TRUE;               //退出应用程序
        break;

    case E_RCV_DSTDChannel:
```

```
            switch(pEvent->dwParam)
            {
            case DST_AUDIO_CHG:
                ……                          //do something
                break;
            case DST_MSG_CHG:
                ……                          //do something
                break;
            default:
                break;
            }
        default:
            break;
        }
        return 1;
}
```

1.14.2 输出事件的数据结构

SynCTI驱动程序在输出事件时，可以使用 2 种数据结构：MESSAGE_INFO和SSM_EVENT。MESSAGE_INFO中包含了常用的参数，适用于CTI系列板卡和REC系列板卡；SSM_EVENT是对MESSAGE_INFO的扩展，能提供更多的事件信息，通常用于DST系列板卡，但要求SynCTI驱动程序的版本为 4.7.3.0 或更高。配置项EventInterfaceType用于选择输出事件的数据结构。

注 意 ： MESSAGE_INFO 适 用 于 函 数 SsmWaitForEvent 和 SsmGetEvent ， SSM_EVENT 适 用 于 SsmWaitForEventA和SsmGetEventA。

1.14.2.1 MESSAGE_INFO

MESSAGE_INFO 结构的声明为：

```
typedef struct    _MESSAGE_INFO
{
    WORD     wEvent;                //事件编码
int          nReference;            //参考值
DWORD    dwParam;                   //输出参数
}MESSAGE_INFO, *PMESSAGE_INFO;
```

- **wEvent**

wEvent 为事件编码。SynCTI 驱动程序抛出的事件大致分为下列几种：

- ✧ E_CHG_**xxxx**： 驱动程序内部的某个状态或计数器发生变化
- ✧ E_PROC_**xxxx**：应用程序提交的某个任务有进展
- ✧ E_SYS_**xxxx**： 驱动程序检测到某个事件发生
- ✧ E_RCV_**xxxx**： 驱动程序从对端交换机收到某个消息或事件

其中，xxxx 表示事件的标志符。

SynCTI 驱动程序将内部产生的事件分为 2 类：常用事件和非常用事件。

全部事件的编码及其特性如下表所示：

事件编码	shpa3api.h 中的宏	常用事件	描述
0x0000	E_PROC_Recognize	√	语音识别结束
0x0001	E_CHG_ISDNStatus	—	ISDN：ISDN 协议的 LAPD 层状态发生变化
0x0002	E_RCV_Ss7Msu	—	SS7：从 SS7 服务器收到新的消息（MSU）
0x0003	E_CHG_Mtp3State	—	SS7：SS7 服务器 Mtp3 层的状态发生变化，通常用于指示到某个 DPC 的路由是否可用
0x0004	保留		
0x0005	E_CHG_FaxPages	√	传真通道：驱动程序完成一页传真的接收或发送

0x0006	E_PROC_FaxEnd	√	传真通道：驱动程序完成传真的发送或接收任务
0x0007	E_CHG_PcmLinkStatus	—	数字中继线的同步状态发生变化
0x0008	E_CHG_LineVoltage	—	模拟电话线路上的电压发生变化
0x0009	E_RCV_CAS	—	SS1 通道：对端交换机的 ABCD 信令码发生变化
0x000A	E_RCV_R2	—	SS1 通道：收到对端交换机的 R2 信号
0x000B	E_PROC_WaitDTMF	√	WaitDTMF 任务完成，任务通过函数 SsmSetWaitDtmf 、 SsmSetWaitDtmfEx 或 SsmSetWaitDtmfExA提交
0x000C	E_CHG_RcvDTMF	√	DTMF 检测器：收到一个 DTMF 字符
0x000D	E_PROC_SendDTMF	√	DTMF发生器：发送DTMF的任务完成，发送DTMF任务由函数SsmTxDtmf启动
0x000E	E_PROC_SendFlash	√	发送闪断信号的任务完成
0x000F	E_PROC_PlayEnd	√	放音操作：放音任务结束。放音任务可以由下列函数之一启动： ✧ SsmPlayFile ✧ SsmPlayIndexString ✧ SsmPlayIndexList ✧ SsmPlayFileList ✧ SsmPlayMem ✧ SsmPlayMemList ✧ SsmPlayMemBlock
0x0010	E_PROC_PlayFile	√	放音操作：文件放音进程指示
0x0011	E_PROC_PlayFileList	√	放音操作：驱动程序完成文件序列中一个文件的播放
0x0012	E_PROC_PlayMem	√	放音操作：单个内存播放任务的进展指示
0x0013	E_PROC_RecordEnd	√	录音操作：录音任务终止
0x0014	E_PROC_RecordFile	√	录音操作：文件录音任务的进展指示
0x0015	E_PROC_RecordMem	√	录音操作：内存录音任务的进展指示
0x0016	E_PROC_SendFSK	√	FSK 数据发送器完成全部数据的发送
0x0017	E_PROC_RcvFSK	√	RcvFSK 任务结束
0x0018	E_CHG_ChState	√	状态机：通道状态发生变化
0x0019	E_PROC_AutoDial	√	状态机：AutoDial 任务有进展
0x001A	E_CHG_RemoteChBlock	—	TUP/ISUP 通道：闭塞对端通道的操作完成
0x001B	E_CHG_RemotePCMBlock	—	TUP/ISUP 通道：闭塞对端 PCM 的操作完成
0x001C	E_SYS_ActualPickup	—	模拟中继线通道：摘机命令执行完毕
0x001D	E_CHG_RingFlag	—	模拟中继线通道/模拟电话线录音通道：铃流信号的电平发生变化
0x001E	E_CHG_RingCount	√	模拟中继线通道：铃流信号检测器中信号周期的计数器发生变化
0x001F	E_CHG_CIDExBuf	√	DTMF 检测器：主叫号码扩展缓冲区的长度发生变化
0x0020	E_CHG_RxPhoNumBuf	√	DTMF 检测器：收到新的被叫号码
0x0021	E_CHG_PolarRvrsCount	—	模拟中继线通道：在线路上检测到一次极性反转
0x0022	E_SYS_RemotePickup	—	模拟中继线通道：增强的远端摘机检测器检测到被叫用户摘机
0x0023	E_CHG_FlashCount	√	坐席通道：在电话机上检测到一次闪断操作
0x0024	E_CHG_HookState	√	坐席通道：话机上检测到摘机或挂机动作
0x0025	E_CHG_ToneAnalyze	√	信号音检测器：分析结果发生变化
0x0026	E_OverallEnergy	—	信号音检测器：线路上的全频能量发生变化
0x0027	E_CHG_OvrlEnrgLevel	√	信号音检测器：全频能量标识输出事件
0x0028	E_CHG_BusyTone	√	信号音检测器：呼叫进程音检测器检测到的忙音周期的个数发生变化
0x0029	E_CHG_BusyToneEx	—	信号音检测器：采用背靠背忙音检测算法检测到的忙音信号
0x002A	E_CHG_VocFxFlag	√	信号音检测器：单音频信号音的电平发生变化，通常用于传真信号音的检测
0x002B	E_CHG_ToneValue	—	信号音检测器：信号音的电平发生变化
0x002C	E_CHG_RingEchoToneTime	—	信号音检测器：回铃音计数器的值发生变化
0x002D	E_CHG_PeakFrq	—	信号音检测器：峰值频率发生变化
0x002E	E_SYS_BargeIn	√	Barge In 检测器：检测结果发生变化
0x002F	E_SYS_NoSound	√	信号音检测器：线路上保持静默
0x0030	E_SYS_TIMEOUT	√	全局定时器：由函数SsmStartTimer启动的定时器发生溢出
0x0031	E_CHG_SpyState	—	DTP 系列：监控电路的状态发生变化
0x0032	保留		
0x003c	E_CHG_CICRxPhoNumBuf	—	SS7 虚电路：收到新的被叫号码
0x003d	E_CHG_CICState	—	SS7 虚电路：电路状态发生变化
0x003e	E_PROC_CICAutoDial	—	SS7 虚电路：ShgAutoDial 任务有进展
0x003f	E_RCV_Ss7IsupUtuinf	—	SS7：收到 USR 消息事件
0x0040	E_CHG_Mtp2Status	√	SS7 信令链路：信令链路的状态发生变化
0x0041	E_RCV_DSTDChannel	√	DST 系列：D 通道事件

➢ **nReference 和 dwParam**

参数 nReference 和 dwParam 的物理含义与参数 wEvent 的值有关，如下表所示：

事件类型	nReference	dwParam
E_PROC_Recognize	通道的逻辑编号	输出识别结果，取值范围为

		1：取得识别结果(Accept) 2：未听懂(Reject) 3：未听清(Failed) 4：未听到(Silence)
E_CHG_ISDNStatus	数字中继线的逻辑编号	不使用。函数SsmISDNGetStatus可用来获取具体的输出参数
E_RCV_Ss7Msu	不使用	不使用。函数SsmGetSs7Msu用于取出消息
E_CHG_Mtp3State	目的信令点(DPC)的编号	输出本端到指定 DPC 的路由上的 SS7 信令服务是否可用的信息，具体为： 1：可用； 0：不可用。
E_CHG_FaxPages	通道的逻辑编号	输出驱动程序完成发送或接收的页面总数
E_PROC_FaxEnd	通道的逻辑编号	输出传真发送或接收任务的进展情况，具体为： 0：传真任务尚未完成，包括握手过程以及传真接收或发送过程； 1：传真任务已经完成，通道回到空闲状态； 2：传真接收或发送过程中发生错误，或者该任务被应用程序调用函数SsmFaxStop终止，通道回到空闲状态； 3：已经完成全部传真数据的发送或接收，开始与对端协商断开连接
E_CHG_PcmLinkStatus	数字中继线的逻辑编号	输出数字中继线的状态值。详细内容请参见函数SsmGetPcmLinkStatus中pwPcmLinkStatus参数的说明。
E_CHG_LineVoltage	通道的逻辑编号	输出模拟电话线上的电压值，单位为伏特。
E_RCV_CAS	通道的逻辑编号	输出收到的 ABCD 信令码的最新值，低 4 位比特有效：Bit3 Bit2 Bit1 Bit0： 对应于 ABCD 码 其余比特： 全部为 0。
E_RCV_R2	通道的逻辑编号	输出最新接收到的 R2 信号，取值范围为 0： R2 信号消失 1~15： 前向 R2 信号的值 1~6： 后向 R2 信号的值
E_PROC_WaitDTMF	通道的逻辑编号	输出 WaitDTMF 任务的终止原因，具体取值为 1： 任务超时 2： 接收到指定的结束字符 3： 接收到指定长度的 DTMF 字符串 函数SsmChkWaitDtmf可用于获取更多的信息。
E_CHG_RcvDTMF	通道的逻辑编号	高 16 比特为收到的DTMF字符的总数，低 16 比特为最新收到的字符，相当于连续调用SsmGetRxDtmfLen和SsmGetLastDtmf函数。
E_PROC_SendDTMF	通道的逻辑编号	=0：完成缓冲区全部 DTMF 字符的发送 =1：发送任务被应用程序终止
E_PROC_SendFlash	通道的逻辑编号	不使用。
E_PROC_PlayEnd	通道的逻辑编号	输出放音任务终止的原因，具体为 1：全部语音数据播放完毕； 2：因收到 DTMF 字符而终止； 3：因检测到 Barge in 而终止 4：因检测到对端用户的挂机动作而终止 5：被应用程序终止 6：文件播放任务被暂停 7：因操作总线而终止 8：文件播放任务因网络故障而终止 更多信息请参见函数SsmCheckPlay的说明。
E_PROC_PlayFile	通道的逻辑编号	输出下列参数之一： ✧ 完成播放的百分比 ✧ 完成播放的时间长度 ✧ 完成播放的字节总数 ✧ 尚未完成播放的字节总数 具体输出哪一个，由SsmSetEvent函数决定，缺省值为输出放音时间。
E_PROC_PlayFileList	通道的逻辑编号	正在播放的语音文件在文件序列中的索引值（从 0 开始编号）。
E_PROC_PlayMem	通道的逻辑编号	输出下列参数之一： -1： 驱动程序的放音指针越过缓冲区的中间位置 -2： 驱动程序的放音指针越过缓冲区尾部后，返回到缓冲区头部 其它值：驱动程序的放音指针在缓冲区中的偏移量，单位为字节。
E_PROC_RecordEnd	通道的逻辑编号	录音任务的终止原因，具体为 1：被应用程序终止； 2：因检测到 DTMF 字符而终止； 3：因检测到对端用户的挂机动作而终止； 4：因录制的数据到达指定长度或时间而终止； 5：文件录音被暂停； 6：将录音数据写入到文件失败。
E_PROC_RecordFile	通道的逻辑编号	输出下列参数之一： ✧ 录音时间 ✧ 完成录音的字节总数

		具体输出哪一个，由SsmSetEvent函数决定，缺省值为输出录音时间。
E_PROC_RecordMem	通道的逻辑编号	输出下列参数之一： -1： 驱动程序内部的录音指针越过内存缓冲区的 1/2 -2： 驱动程序内部的录音指针越过内存缓冲区的尾部 其它值：录音指针偏移量。
E_PROC_SendFSK	通道的逻辑编号	0-保留未使用
E_PROC_RcvFSK	通道的逻辑编号	FSK 数据接收器的停止原因： 1：超时，接收失败； 2：收到指定的结束特征字节而结束； 3：收到指定长度 FSK 数据而结束； 4：收到指定格式的数据而结束； 5：应用程序调用函数SsmStopRcvFSK中断。 应用程序得到本事件后，可以调用函数SsmGetRcvFSK取得收到的FSK数据。
E_CHG_ChState	通道的逻辑编号	高 16 位比特表示通道的上一个状态值； 低 16 位比特表示通道的新状态值。 有关通道状态的取值及其描述请参见函数SsmGetChState的说明。
E_PROC_AutoDial	通道的逻辑编号	AutoDial任务进展值，参数的具体含义请参见函数SsmChkAutoDial的说明。
E_CHG_RemoteChBlock	通道的逻辑编号	闭塞/解闭塞对端电路操作的进展情况，具体值为： 0：已经成功解除由本端导致的闭塞； 1：已经成功闭塞对端； 2：正在等待对端交换机的闭塞证实信号； 3：正在等待对端交换机的解除闭塞证实信号。
E_CHG_RemotePCMBlock	数字中继线的逻辑编号	闭塞/解闭塞对端数字中继线的进展情况，参数返回低 16 位为： 0：已经成功解除由本端导致的闭塞； 1：已经成功闭塞对端； 2：正在等待对端交换机的闭塞证实信号； 3：正在等待对端交换机的解除闭塞证实信号。 高 16 位为闭塞的模式。
E_SYS_ActualPickup	通道的逻辑编号	0（保留，未使用）
E_CHG_RingFlag	通道的逻辑编号	铃流信号的电平发生变化： Bit31： 铃流电平标志， 1 ：有铃流； 0：无铃流； Bit30~Bit0：上一个铃流电平的保持时间，单位：毫秒。
E_CHG_RingCount	通道的逻辑编号	铃流信号周期的个数
E_CHG_CIDExBuf	通道的逻辑编号	主叫号码扩展缓冲区中的字符总数。调用SsmGetCallerIdEx函数可以获取字符内容。
E_CHG_RxPhoNumBuf	通道的逻辑编号	被叫号码缓冲区中的字符总数。调用SsmGetPhoNumStr函数可取得被叫号码的具体内容。
E_CHG_PolarRvrsCount	通道的逻辑编号	驱动程序内部保存极性反转的计数器的值
E_SYS_RemotePickup	通道的逻辑编号	0（保留，未使用）
E_CHG_FlashCount	通道的逻辑编号	检测到的闪断的次数
E_CHG_HookState	通道的逻辑编号	用户对话机的动作： 0：挂机。 1：摘机。
E_CHG_ToneAnalyze	通道的逻辑编号	信号音分析器结果： 信号音分析器忙音结果： 高 16 比特：呼叫进程音分析器的编号，0：第一组，1：第二组； 低 16 比特：检测到的信号音，具体取值为： 1： 检测到拨号音； 2： 检测到忙音； 3： 检测到回铃音； 4： 检测到回铃音后，线路保持静默； 5： 检测到无声； 6： 检测到说话声，用于自动拨号时检测被叫用户是否应答（摘机）； 7： 检测到F1 频率的信号音，F1 频率参数由函数SsmSetVoiceFxPara设置。用于自动拨号时检测被叫用户的应答类型； 8： 检测到F2 频率的信号音，F2 频率参数由函数SsmSetVoiceFxPara设置。用于自动拨号时检测被叫用户的应答类型； 9： 检测到用户指定的信号音类型。
E_OverallEnergy	通道的逻辑编号	信号音分析器全频能量值。
E_CHG_OvrlEnrgLevel	通道的逻辑编号	信号音分析器全频能量标识值，低 16 比特： Bit15 (信号音标志位)：0/1 无/有信号音。 Bit14 (函数操作状态位)：0 保留 Bit13-0 (持续时间)：有/无信号音的持续时间。
E_CHG_BusyTone	通道的逻辑编号	信号音检测器的忙音检测结果： 高 16 比特：指示信号音分析器，0 第一组，1 第二组； 低 16 比特：忙音周期的个数。

E_CHG_BusyToneEx	通道的逻辑编号	信号音检测器采用背靠背忙音检测算法检测到的忙音信号： 0：忙音信号消失； 1：检测到忙音信号。
E_CHG_VocFxFlag	通道的逻辑编号	传真信号音的类型，具体为： 1：检测到 F1； 2：检测到 F2；
E_CHG_ToneValue	通道的逻辑编号	信号音分析器：信号音电平变化值。 低 16 比特有效： Bit15(信号音标志位) ： 0/1 有/无信号音。 Bit14(函数操作状态位)：0/1 操作成功/操作失败。 Bit13-0(持续时间)：有/无信号音的持续时间。
E_CHG_RingEchoToneTime	通道的逻辑编号	信号音分析器：回铃音时间，以一个回铃音周期变化，单位秒。
E_CHG_PeakFrq	通道的逻辑编号	信号音分析器：峰值频率
E_SYS_BargeIn	通道的逻辑编号	BargeIn 检测器的检测结果发生变化： 0：Barge In 消失； 1：检测到 BargeIn。
E_SYS_NoSound	通道的逻辑编号	不使用
E_SYS_TIMEOUT	定时器的编号	不使用
E_CHG_SpyState	SpyCic 的逻辑编号	高 16 位比特表示 SpyCic 的上一个状态值； 低 16 位比特表示 SpyCic 的新状态值。 有关SpyCic状态的取值及其描述请参见函数SpyGetState的说明
E_RCV_DTR_AUDIO	通道的逻辑编号	=1 时为语音通道打开消息，=0 时为语音通道关闭消息
E_RCV_DTR_HOOK	通道的逻辑编号	=1 时为通道摘机消息，=0 时为通道挂机消息。
E_RCV_DTR_LAMP	通道的逻辑编号	HWORD wState：灯的状态，LWORD wPos：灯的位置)，状态和位置信息请参考附录的“数字电话录音”节。
E_RCV_DTR_FKEY	通道的逻辑编号	DWORD dwFkeyValue，功能键键值，请参考附录的“数字电话录音”节。
E_RCV_DTR_DKEY	通道的逻辑编号	DWORD dwDKeyValue，拨号键键值，请参考附录的“数字电话录音”节。
E_RCV_DTR_VOX	通道的逻辑编号	=1 表示 VOX 打开，=0 表示 VOX 关闭。
E_RCV_DTR_DISPLAY	通道的逻辑编号	显示屏显示内容长度
E_RCV_DTR_DIRECTION	通道的逻辑编号	=1 表示呼出，=0 呼入
E_RCV_DTR_RING	通道的逻辑编号	=1 表示振铃开始，=0 表示振铃结束
E_CHG_CICRxPhoNumBuf	虚电路编号	被叫号码长度。 应用程序得到该事件后，要调用 ShgGetPhoNumStr 函数取得被叫号码。
E_CHG_CICState	虚电路编号	电路状态值。详细内容参见 ShgGetChState 的说明。
E_PROC_CICAutoDial	虚电路编号	ShgAutoDial 任务进展值，具体为： 1：DIAL_ECHOTONE = 2，发送完被叫号码后检测到了回铃音 3：DIAL_BUSYTONE= 4，被叫用户忙，自动拨号结束 6：DIAL_VOICE= 7，被叫用户摘机，自动拨号结束 9：DIAL_NOANSWER= 10，无人接听，自动拨号失败 10：DIAL_FAILURE= 11，自动拨号失败 11：DIAL_INVALID_PHONUM = 12，空号，自动拨号结束
E_RCV_DSTDChannel	通道的逻辑编号	子事件码，如 DST_AUDIO_CHG、DST_MSG_CHG 等以“DST_”开头的子事件码，具体事件码的含义请参见《SynCTI 程序员手册-D 信道事件用户手册》
E_RCV_Ss7IsupUtuinf	应用程序通道号	用户至用户信息，其所指向的空间中，第一个字节是数据长度（unsigned int），随后是数据（unsigned char[]）。
E_CHG_Mtp2Status	SS7 信令链路号	SS7 信令链路状态值，具体为： 1：业务中断 2：初始定位 3：已定位/准备好 4：已定位/未准备好 5：业务开通 6：处理机故障

1.14.2.2 SSM_EVENT

SSM_EVENT 结构的声明为：

```
typedef struct   _SSM_EVENT
{
    WORD    wEventCode;        //事件编码
    int     nReference;        //参考值
    DWORD   dwParam;           //输出参数

    DWORD    dwUser;           //用户参数
    DWORD   dwSubReason;       //子原因值
```

```
    DWORD   dwXtraInfo;          //扩展信息值

    PVOID   pvBuffer;            //事件信息缓冲指针
    DWORD   dwBufferLength;      //事件信息缓冲大小

    DWORD   dwDataLength;        //事件信息大小
    DWORD   dwEventFlag;         //事件标记

    DWORD       dwReserved1;     //保留参数 1
    LONGLONG    dwReserved2;     //保留参数 2
}SSM_EVENT, *PSSM_EVENT;
```

结构中的参数说明如下：

- **wEventCode、nReference、dwParam**

  这 3 个参数的含义与 MESSAGE_INFO 结构中的 wEvent、nReference 和 dwParam 完全相同。

- **dwUser**

  参数 **dwUser** 的含义与驱动程序输出事件的模式有关，如下表所示：

事件输出模式	dwUser 的含义
事件等待	当事件为用户自定义事件时，dwUser的值等于调用SsmPutUserEventA时的参数pEvent的参数dwUser的值，否则没有意义
事件回调	透明传递应用程序在调用函数SsmSetEvent时参数pEventSet中的dwUser参数值

- **dwSubReason**

  参数 dwSubReason 为子原因值，只有当参数 wEventCode 等于 E_RCV_DSTDChannel 时才有意义。dwSubReason 的具体含义与驱动程序抛出事件时携带的 dwParam 参数有关，更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。

- **dwXtraInfo**

  参数 dwXtraInfo 为扩展信息，只有当参数 wEventCode 等于 E_RCV_DSTDChannel 时才有意义。dwXtraInfo 的具体含义与驱动程序抛出事件时携带的 dwParam 参数有关，更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。

- **pvBuffer、dwBufferLength 、dwDataLength**

  pvBuffer、dwBufferLength、dwDataLength 只在参数 *wEventCode* 为 E_RCV_DSTDChannel 时有意义。具体含义与驱动程序输出事件的模式有关，如下表所示：

事件输出模式	含义
事件等待	pvBuffer为指向事件信息缓冲区的指针。应用程序在调用函数SsmWaitForEventA或SsmGetEventA时，应该为pvBuffer分配不小于 300 个字节的内存空间，而且应该设置dwBufferLength的值为实际分配的内存空间的大小（单位为字节）。函数SsmWaitForEventA或SsmGetEventA返回事件后，pvBuffer中内容的含义与驱动程序抛出事件时携带的dwParam参数有关，更多信息请参见《SynCTI 程序员手册-D信道事件用户手册》中的相关内容。dwDataLength为pvBuffer中数据的实际长度（单位为字节）。
事件回调	pvBuffer 为指向事件信息缓冲区的指针。驱动程序内部产生 E_RCV_DSTDChannel 事件后，在调用应用程序预先注册的回调函数前，会自动为 pvBuffer 分配内存空间，因此，参数 dwBufferLength 没有意义。pvBuffer 中内容的含义与驱动程序抛出事件时携带的 dwParam 参数有关，更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。dwDataLength 为 pvBuffer 中数据的实际长度（单位为字节）。

- **dwEventFlag**

  dwEventFlag 为事件标记。dwEventFlag 的 Bit0 和 Bit2 有效，其余比特保留未用。Bit0 和 Bit2 的含义如下表所示：

比特	含义
Bit0	=0：本事件为驱动程序创建的事件 =1：本事件为应用程序创建的事件，即函数SsmPutUserEventA向驱动程序提交的自定义事件
Bit2	=0：事件信息没有被截断 =1：事件信息被截断 Bit2 只在事件输出模式为事件等待时有意义。只要符合以下两个条件之一 Bit2 将被驱动程序设置为 1： ✧ 应用程序调用函数SsmPutUserEventA时参数pEventSet中的dwDataLength参数值大于 256 ✧ 事件在输出之前临时保存在驱动程序中，应用程序调用函数SsmWaitForEventA时参数 pEventSet中的dwBufferLength参数值小于驱动程序中该事件的参数dwDataLength

➢ **dwReserved1、dwReserved2**

保留参数。

1.14.3 事件的过滤

应用程序可以决定一个SynCTI驱动程序抛出或不抛出一个事件。这项功能是通过函数SsmSetEvent来实现。

当应用程序以wEvent=0xffff作为参数调用函数SsmSetEvent时，驱动程序抛出或不抛出某个事件，取决于配置项DefaultEventOutput的设置，详细信息请参见该配置项的说明。

1.14.4 使用自定义事件

应用程序在运行过程中，也可能产生自己的事件，称为自定义事件。为了方便应用程序的编程，驱动程序提供了自定义事件的透明传输功能，即应用程序可以将自定义事件缓存到驱动程序内部的事件队列中，再通过驱动程序提供的事件获取函数进行处理。

函数SsmPutUserEvent和SsmPutUserEventA可以用来向驱动程序提交自定义事件。

1.14.5 常见编程平台下的编程要点

本小节介绍 SynCTI for Windows 驱动程序分别在 MS VC/C++、VB、C++ BUILDER、Delphi、PB6.5 以及其它编程平台下的编程要点，和 SynCTI for Linux 驱动程序的 C 语言编程要点。

1.14.5.1 MS VC/C++

SynCTI 驱动程序提供了在 MS VC++编程平台下需要用到下列文件：

✧ shpa3api.h： C/C++头文件

✧ shp_a3.lib： C/C++输入库文件

1.14.5.2 VB

在 Visual Basic 6.0 中，对动态链接库 SHP_A3.DLL 的调用是通过 Shpa3api.bas（由驱动程序提供） 来声明的，请参见 Shpa3api.bas。

在本质上，采用 VB 编写的应用程序也是通过调用 shp_a3.dll 来实现对 Synway 板卡硬件的控制，因此，上一节中有关在 VC 环境下编程的说明，同样适用于 VB，请仔细阅读上一节。在开发应用程序时，需要经常参考 C 语言的示范程序。

在 VB 6.0 中，需要首先对 DLL 中的函数进行声明，然后才能调用相关函数。如何声明 DLL 中的函数，请参考 VB 中的 HELP 文件“Declare statement”。

当 shp_a3.dll 中有新的函数未能包含在 VB 声明文件 Shpa3api.bas 中时，可以参考该函数在 C 语言中的声明，来对需要在 VB 下调用的函数进行声明。例如，在 Shpa3api.h 中声明的一个函数：

```
int WINAPI SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen);
```

在 Shpa3api.bas 中声明为：

```
Declare Function SsmPlayFile Lib "shp_a3.dll" (ByVal ch As Long, ByVal pszFileName As String,
                                               ByVal  nFormat  as Long, ByVal dwStartPos as Long,
                                               ByVal dwLen As Long) As Long
```

由于 VC 采用 32 位方式进行编译，类型 BOOL 为 32 位、WORD 为 16 位、DWORD 为 32 位，因此，在 VB 中，相应地使用 Long（32 位）、 Integer（16 位）、 Long（32 位）。另外，需要注意的是，**所有参数和返回值的字节长度要相等。**

同在 9x/NT 下用 C 语言编程一样，VB 对板卡的操作也包括三部分：

- 初始化代码，包括初始化驱动程序、获取通道总数、通道类型等，在 Form_Load 中运行。
- 事务处理代码，在 TIMER 中运行，或放在 Form_Activate 函数里，具体用法请参见相关演示程序
- 卸载代码，包括卸载驱动程序、关闭板卡等，在 Form_Unload 中运行。

注意：

- Form_Activate 是一个死循环，如果将处理部分放在这里，要调用 DoEvents，以便让其它程序能正常工作。
- 由于 VB 是解释性语言，在调用 DLL 时，如果某函数在 C 语言中的参数为字符串，并且在函数调用完毕后，需要通过该参数返回数据，如 SsmGetLastErrMsg、SsmGetDtmfDStr 等函数，相关参数必须预先分配空间。例如：

  ```
  Dim ErrStr As String * 200
  SsmGetLastErrMsg( ErrStr)
  ```

- 在 Windows NT 环境下，如果将初始化代码和主处理程序代码分别放在 Form_Load、Timer 中，必须保证在成功调用 SsmStartCti 后再执行 Timer 中的代码。具体方法：使用一个全局变量，在 Form_Load 的尾部将其置为 TRUE；在 Timer 中根据该变量的值来决定是否开始工作。也可以在 Form_Load 的开始部分调用 Timer.Enable=False，在最后调用 Timer.Enable = True；在 Form_Unload 时，先调用 Timer.Enable=False，再调用 SsmCloseCti。

1.14.5.3 C++ BUILDER

驱动程序提供的 Shpa3api.h 可以直接使用。

shp_a3.lib 是在 VC 环境编译的，不能直接用于 C++ BUILDER 工程文件的连接。C++ BUILDER 提供了一个从 DLL 中提取输入库 LIB 的工具软件 IMPLIB.EXE，可用该软件生成一个 LIB 用于开发。IMPLIB.EXE 的用法，请参考其随机的帮助文件。IMPLIB.EXE 的使用方法：

```
cd \windows\system
implib shp_a3.lib shp_a3.dll
```

将生成的 shp_a3.lib 拷贝到应用程序的开发目录下即可。

注意事项：

- **如果硬盘上装有两套 Borland 公司的 C 语言开发环境，要保证使用 C++ BUILDER 所附带的 IMPLIB.EXE。否则会发生错误！**
- **SynCTI Ver. 3.3.1 及以后版本将 Windows 和 Linux 的 Shpa3api.h 头文件合并为一个，因此，在使用 BCB 编译时，需要在#include "shpa3api.h"语句的前一行加上#define WIN32，即：**

  ```
  #define WIN32
  #include "shpa3api.h"
  ```

1.14.5.4 Delphi

在 Delphi 编译环境下，对 shp_a3.dll 的调用是通过 Shpa3api.pas 来声明的，详细情况请参见 Shpa3api.pas。

与 VB6.0 相同，在 Delphi 中也需要首先对 DLL 中的函数进行声明，然后才能对该函数进行调用。关于如何声明 DLL 中的函数，请参考 Delphi 提供的 HELP 文件“external declarations”。如果 shp_a3.dll 中有某个函数没有包含在文件 Shpa3api.pas 中，可以参考该函数在 C 语言中的声明，自己来声明 Delphi 下的函数调用。例如，在 Shpa3api.h 中有下面的函数：

int WINAPI SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen);

在 Delphi 中可以声明为：

function SsmPlayFile (ch: Integer; pszFileName: PCHAR; nFormat:Integer; dwStartPos: DWORD;dwLen:DWORD) : Integer; stdcall; external 'Shp_a3.dll'。

注意：

- 声明的最后必须加 stdcall; external 'Shp_a3.dll'。
- C 语言中的 char*应声明为 pchar。

1.14.5.5 PB6.5

在 Power Builder 编译环境下，对 shp_a3.dll 的调用是通过在“Decalre”菜单中的“Global External Function”来声明的。如果 shp_a3.dll 中有某个函数没有包含在声明中时，可以参考该函数在 C 语言中的声明，自行声明在 PB6.5 下的函数调用。例如，在 Shpa3api.h 中的函数声明为：

int WINAPI SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen);

在 Power Builder 环境下：

Function Long SsmPlayFile (Long ch,String pszFileName,Long nFormat, UnsignedLong dwStartPos,UnsignedLong dwLen) Library "shp_a3.dll"

注意： PB6.5 编译环境下的参数类型与 VB 相拟，请参阅“VB 编程平台编程要点”小节。

1.14.5.6 其它

在 Windows 95/NT 的 32 位编程环境中，许多编程语言都可以用各种方式来调用 DLL 动态库，因此，同样可以实现对动态库 SHP_A3.DLL 的调用。由于编程语言种类繁多，本手册无法一一描述。请仔细阅读你所用的编程语言的说明。

1.14.5.7 Linux 操作系统

C/C++应用程序需要用到下列文件（由驱动程序提供）：

- shpa3api.h： C/C++头文件
- libshpa3.so： 驱动程序调用接口库

在 Linux 下用 C / C++语言调用 SynCTI 驱动程序时，一般包括三部分：

- 初始化代码，包括初始化卡、得到通道数和通道类型等等，在应用程序启动时调用一次。
- 事务处理代码，向驱动程序发布命令，对驱动程序抛出的事件进行处理等。具体用法请参见示范程序。

➢ 结束部分，包括释放语音卡、释放驱动程序，在应用程序退出时调用。

1.14.6 配套演示程序（CTI 系列）

为便于开发人员迅速理解一些函数或功能的使用方法，Synway 提供了大量的演示程序，如下表所示：

名称	描述	编程平台					
		Windows					Linux
		VB	VC	C#	PB	DELPHI	gcc
Call	演示如何处理电话呼入。	√	√	—	√	√	√
Dial	演示如何实现电话呼出。	√	√	—	√	√	√
Conference Demo	演示电话会议的编程方法，适用于坐席通道、模拟外线通道、数字中继通道。更多内容请参见 Demo 文件夹下的 DemoConferenceUserManual.pdf（英文）或 DemoConferenceUserManual_cn.pdf（中文）。	—	√	—	—	—	—
Fax	演示如何用 Synway 板卡实现一个简易的传真服务器。详见 Demo 文件夹下的 DemoFAXUserManual.pdf（英文）或 DemoFAXUserManual_cn.pdf（中文）。	√	√	√	—	√	√
Pbx	用 Synway 板卡模拟一台小型交换机。详见 DemoPbxUserManual.pdf（英文）或 DemoPbxUserManual_cn.pdf（中文）。	—	√	√	—	—	—
	数字中继卡调试平台。用于调试基于数字中继卡的应用程序，具有简单的交换机功能。	—	√	—	—	—	—
FSK	演示如何利用 Synway 板卡的 FSK 数据传输功能实现接收和发送数据。	—	√	—	—	—	—
Ss7Server	实现 SS7 信令服务器功能的演示程序。	—	√	—	—	—	—
Test	演示 API 接口函数的使用方法。	—	√	—	—	—	√
多线程VC	演示如何使用多线程方式对 Synway 板卡进行编程	—	√	—	—	—	—
事件驱动 demo	演示如何使用 SynCTI 驱动程序抛出的事件进行编程	—	√	—	—	—	—

1.14.7 配套演示程序（REC 系列）

为便于开发人员迅速理解一些函数或功能的使用方法，Synway 提供了大量的演示程序，如下表所示：

演示程序名称	描述	编程平台			
		Windows			Linux
		VB	VC	C#	gcc
Recorder-ATP	演示如何使用ATP系列录音卡实现对模拟电话线路进行侦听和录音。详细说明请查阅 DemoATPUserManual.pdf（英文）或DemoATPUserManual_cn.pdf（中文）。	—	√	√	√
Recorder-DST	演示如何使用 DST 系列板卡对数字电话用户线进行高阻并线录音、监听等功能。详见 DemoDSTUserManual.pdf（英文）或 DemoDSTUserManual_cn.pdf（中文）。	√	√	—	—
Recorder-DTP	演示如何使用DTP系列板卡实现对数字中继线（E1）进行监控录音的功能。	√	√	—	—
RecPlay UseMemBlock	演示如何使用乒乓内存录音方式。	—	√	—	—

1.14.8 使用驱动程序提供的定时器

驱动程序向应用程序提供了 2 种方式的定时器：全局定时器和通道定时器。

◆ 全局定时器

驱动程序内部设置了 128 个系统定时器，使用Syn板卡的硬件中断信号作为计时脉冲，精度优于操作系统提供的定时器。函数SsmStartTimer用于向驱动程序申请一个定时器，并设置其工作参数。每当定时器发生溢出时，驱动程序会向应用程序抛出E_SYS_TIMEOUT事件。函数SsmStopTimer用于终止一个定时器。

◆ 通道定时器

通道定时器适用于轮询方式的编程模式。通道定时器启动后，驱动程序只是将当前时刻保存到变量中，并不会在驱动程序内部判断定时器是否溢出。只有当应用程序调用ElapseTime函数时，驱动程序才会返回定时器的保持时间。驱动程序为每个通道分配了 10 个定时器，通过函数StartTimer启动。

1.14.9 使用驱动程序提供的调试功能

SynCTI 驱动程序带有调式功能，通常用于开发阶段、测试阶段及试运行的应用系统，具有 API 调试功能和事件消息调式功能。当应用程序调用 API 函数发生错误，或者事件消息出错时，或驱动程序内部发生某种差错时，如果相应的调试信息输出开关为开启状态，驱动程序会输出错误信息，以帮助开发人员迅速确定出错位置。

需要调试应用程序时，可使用下述配置项：

配置项	描述
EnableDebugAPI	设置是否输出应用程序调用 API 函数的调试信息
SetEventRange	在设置 EnableDebugAPI 的条件下，设置事件输出写 LOG 的事件范围
SetChRange	在设置 EnableDebugAPI 的条件下，设置事件输出写 LOG 的通道范围

如果应用程序采用轮询方式，需要在定时器中不停地调用一些查询函数，从而导致大量的函数调用信息输出。下列配置项可以禁止一些常用的查询类型的函数输出调用信息：

配置项	描述
Mask_SsmGetNoSoundTime	控制是否输出SsmGetNoSoundTime函数的调用信息
Mask_SsmGetChStateKeepTime	控制是否输出SsmGetChStateKeepTime函数的调用信息
Mask_SsmGetPlayedTime	控制是否输出SsmGetPlayedTime函数的调用信息
Mask_SsmGetPlayedPercentage	控制是否输出SsmGetPlayedPercentage函数的调用信息
Mask_SsmGetPlayOffset	控制是否输出SsmGetPlayOffset函数的调用信息
Mask_SsmGetRecTime	控制是否输出SsmGetRecTime函数的调用信息
Mask_SsmGetRecOffset	控制是否输出SsmGetRecOffset函数的调用信息

驱动程序输出的 API 调试信息需要借助于第三方的工具软件 Debugview.exe 来显示和观察。该工具软件支持 Windows 95、 Windows 98、Windows Me、Windows NT 4、Windows 2000、Whistler Beta 1、Beta 2、以及 RC 1。

DebugView.exe是一款绿色软件，不需要安装，直接将下载的ZIP文件解压缩后即可直接运行。Debugview.exe 可以从以下站点免费下载：http://www.sysinternals.com/ntw2k/freeware/debugview.shtml。

1.15 SHT 系列板卡（CTI 系列）

1.15.1 SHT 系列板卡简介

SHT 系列板卡支持的功能如下表所示：

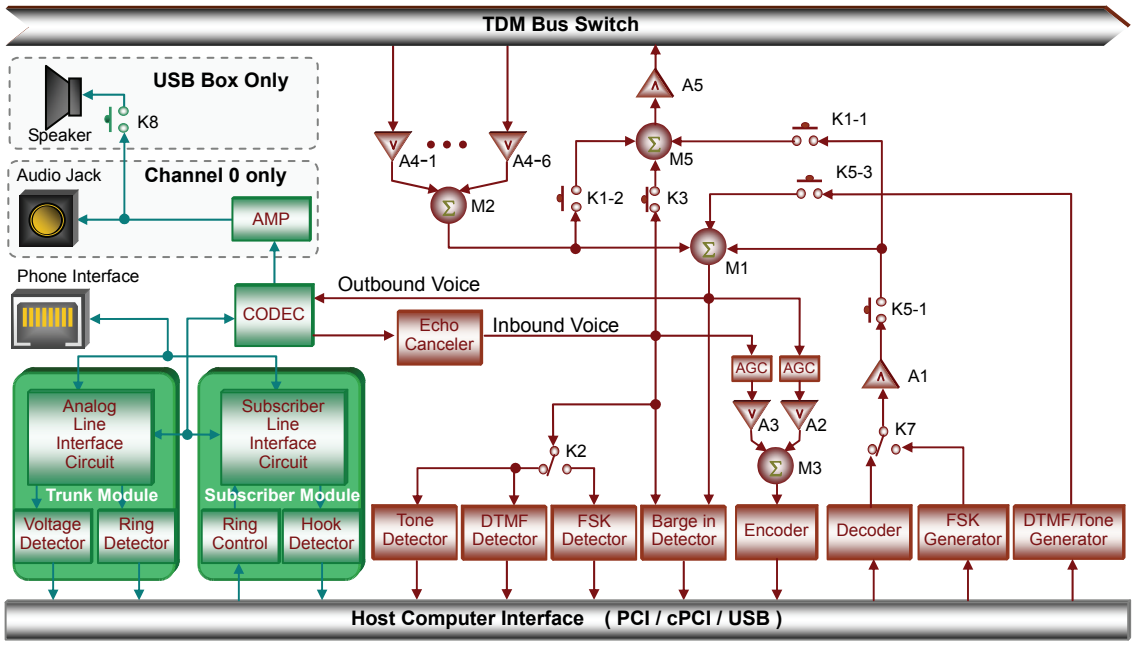
Board Model	Form Factor	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	Voltage Detector	Audio Socket	FSK	TDM Bus	Max Fax Resource
SHT-2A/USB	USB	2	√	√	√	√	√	N/A	N/A
SHT-4A/USB	USB	4	√	√	√	√	√	N/A	N/A
SHT-2B/USB	USB	2	√	√	√	√	√	N/A	N/A
SHT-4B/USB	USB	4	√	√	√	√	√	N/A	N/A
SHT-8A/PCI	PCI	8	√	√	√	√	√	N/A	N/A
SHT-8B/PCI	PCI	8	√	√	√	√	√	N/A	N/A

SHT-8B/PCI/FAX	PCI	8	√	√	√	√	√	N/A	4
SHT-8C/PCI/FAX	PCI	8	√	√	√	√	√	N/A	4
SHT-8C/PCI/EC	PCI	8	√	√	√	√	√	N/A	N/A
SHT-16A-CT/PCI	PCI	16	√	√	√	√	√	H.100	N/A
SHT-16B-CT/PCI	PCI	16	√	√	√	√	√	H.100	N/A
SHT-16B-CT/PCI/FAX	PCI	16	√	√	√	√	√	H.100	4
SHT-16C-CT/PCI/FAX	PCI	16	√	√	√	√	√	H.100	4
SHT-16B-CT/PCI/MP3	PCI	16	√	√	√	√	√	H.100	N/A
SHT-120A-CT/PCI	PCI	120	√	√	N/A	N/A	N/A	N/A	N/A
SHT-16B-CT/cPCI	cPCI	16	√	√	√	√	√	H.110	N/A
SHT-16B-CT/cPCI/MP3	cPCI	16	√	√	√	√	√	H.110	N/A
SHT-16B-CT/cPCI/FAX	cPCI	16	√	√	√	√	√	H.110	8
SHT-120A-CT/cPCI	cPCI	120	√	√	N/A	N/A	N/A	N/A	N/A

注意：型号为SHT-16B-CT/PCI/FAX和SHT-16B-CT/PCI/MP3 的板卡具有不同的硬件结构，但由于历史的原因，这 2 个型号的板卡上固件中写入的板卡ID号相同，因此，需要对配置项DSP3WORKMODE进行设置，以便使驱动程序能够正确区分这 2 种型号的板卡。

1.15.2 SHT 系列板卡的工作原理框图

SHT 系列板卡的工作原理图如下图所示：



原理框图中的模块及符号说明：

名称	描述
M1	去话混音器
K7	语音播放/FSK数据发送器的切换开关，由驱动程序自动控制。当应用程序调用播放函数时，K7 指向语音编码器模块Decoder；当应用程序调用SsmStartSendFSK函数启动发送FSK数据时，K7 执行FSK Generator。有关语音播放的函数请参见语音播放部分内容。
A1	放音的音量调节器，可以通过配置项DefaultPlayVolume设置，也可通过函数SsmSetPlayVolume或SsmSetPlayGain进行调节，缺省的音量设置为 0。
K5-1	控制放音操作的开关，由驱动程序自动控制通常处于断开状态，只有当应用程序调用播放函数时，驱动程序才会自动闭合K5-1 开关；当放音任务结束时，驱动程序自动断开K5-1。有关语音播放的函数请参

	见语音播放部分内容。
K5-3	信号音发生器、DTMF发生器的信号输出控制开关，由驱动程序自动控制，通常处于断开状态，只有当应用程序调用SsmTxDtmf函数启动DTMF发生器，或者调用SsmSendTone/ SsmSendToneEx函数启动信号音发生器后，驱动程序才会自动闭合K5-3 开关；发生器的任务终止后，K5-3 自动断开。
M2	下总线信号混音器
A4-1 … A4-6	下总线信号进入M2 前的音量调节器，可以通过函数SsmSetListenVlmInConf进行设置，缺省值为 0DB。
M3	录音混音器。 M3 的输出进入编码器（Encoder）模块，然后形成录音数据。
A3	来话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecVolume进行设置，也可以通过配置项DefaultRecordVolume设置，缺省值为 0。
A2	去话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecMixer进行设置，也可以通过配置项DefaultRecordMixerVolume设置，缺省值-7，关闭A2。
M5	上总线混音器。 用于将放音数据、下总线信号混音器 M2 的输出、来话信号进行混音，混音结果上行到 TDM 交换总线。板卡初始化时，会将M5 的输出上行到TDM总线的某个时隙（记为ts）上，函数SsmLinkToBus可以用来改变ts。
A5	M5 进入交换总线前的音量增益，可以通过配置项DefaultSpeakVolume或下列函数进行设置： ✧ SsmTalkWithEx ✧ SsmListenToEx / SsmLinkFromEx / SsmLinkFromAllCh
K3	来话信号是否进入上总线混音器M5 的控制开关，可通过函数SsmSetFlag（携带F_InVoiceToBus参数）或配置项InVoiceToBus进行设置，缺省值为闭合。
K1-2	M2 的输出信号是否重新回到TDM交换总线的控制开关，可以通过函数SsmSetFlag（携带F_MixerResToBus参数）进行设置，缺省值为断开。
K1-1	控制放音数据是否进入混音器M5，进而上行到TDM交换总线上的开关，通常为断开状态，可以通过函数SsmSetPlayDest进行控制。K1-1 只能用于电话会议中给会议室播放背景音。
K2	FSK 检测器与其它检测器之间的切换开关，通常指向其它的检测器，只有在开启 FSK 检测器期间才会指向 FSK 检测器。

➢ **放音操作**

应用程序调用播放函数后，驱动程序自动将 K7 指向语音解码器 Decoder，K5-1 闭合。解码后的语音数据经过音量调节器 A1 进行音量调节后，可以有 2 个去向：

✧ 进入 M1 混音器，与其它信号源混音后形成最终的去话信号；如果是 0 通道，去话信号会同时送到板载扬声器端口；

✧ 在 K1-1 控制下进入混音器 M5，与其它信号源混音后，形成上总线的信号源。只能用于电话会议中给会议室播放背景音。

➢ **录音操作的信号源选择范例**

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

✧ 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);          // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);    // 关闭 A2
```

```
SsmRecToFile(ch1, ……);          // 启动文件录音任务……
```

✧ 范例 2：ch1 与 ch2 建立了双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）＋总线混音器 M2 输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);           // ch1 与 ch2 通过 TDM 交换总线建立双向连接
SsmSetRecVolume(ch1, 2);         // 开启 A3，设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);   // 开启 A2，设置增益为-1*3DB=-3DB
SsmRecToFile(ch1, ……);           // 启动文件录音任务……
```

✧ 范例 3：ch1 加入了一个电话会议室（假设会议室编号为 n），在 ch1 上播放背景音乐，同时通过 ch1 对电话会议进行录音。所有信号源均为正常音量，即增益为 0。实现代码为：

```
SsmJoinConfGroup(n, ch1, ……);    // ch1 加入会议室
SsmSetPlayDest(int ch, 1);       // 闭合 K1-1，将放音数据送到总线上
SsmPlayFile(ch1, ……);            // ch1 启动放音任务，播放背景音乐
SsmSetRecVolume(ch1, 0);         // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, TRUE, 0);    // 开启 A2，设置增益为 0DB
SsmRecToFile(ch1, ……);           // 启动文件录音任务……
```

➢ **板载扬声器输出端口**

SHT 系列板卡的 0 通道具有模拟音频放大电路和板载扬声器输出端口，可直接连接到外置的扬声器或耳机，用来对语音信号进行实时监听。其主要用途为：

✧ 将某个通道的来话信号经TDM交换总线直接送到扬声器进行播放。实现此功能的函数为SsmListenTo或SsmListenToEx。

✧ 调用语音播放的函数将应用程序录制的语音文件直接送到板载扬声器进行播放。有关语音播放函数的详细内容请参见本章中语音播放部分。

原理框图中的AMP模块是模拟音频放大电路的功率放大器，它的增益可以通过函数SsmSetPowerAmpVlm设置。

USB总线的语音盒（型号中带“/USB”标记）的 0 通道除了在板卡上安装了扬声器插座外，还有一个板载扬声器，可以将声音字节播放出来。开关K8 用于选择将 0 通道的输出信号是送到扬声器插座，还是板载扬声器。K8 可以通过函数SsmSetLine0OutTo或配置项USBLine0Output进行设置，缺省为输出到扬声器插座。

注意：将语音信号送到 0 通道上板载扬声器的同时，也会作为去话信号送到线路上。

1.15.3 模拟中继线通道

1.15.3.1 在模拟线路上产生闪断信号

闪断信号是指电话机进入通话后，通过快速拍打叉簧而产生的短时挂机信号，通常用于与交换机配合实现转分机等操作。在话机上拍下叉簧的时间应该足够短，以免交换机将闪断信号误判为挂机信号。

在 SynCTI 驱动程序中，可以有 2 种方法在模拟中继线上产生一次闪断信号：

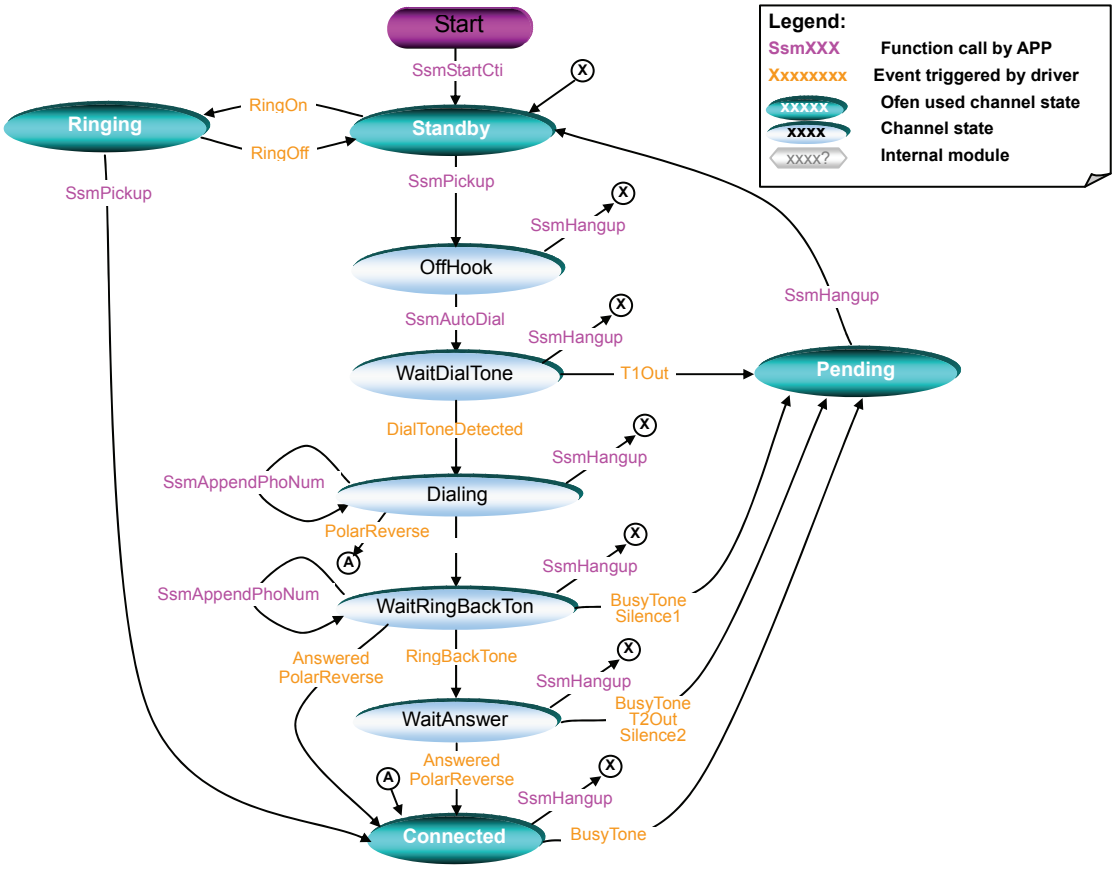
➢ 调用SsmTxFlash函数。可直接指定闪断信号的保持时间

➢ 使用'!'字符作为参数调用函数SsmTxDtmf。此时，产生的闪断信号的保持时间可以通过配置项DefaultTxFlashTime进行设置，缺省值为 500ms

1.15.3.2 检测极性反转信号

有关检测极性反转信号的详细内容请参见本章中“模拟电话线的电压变化”部分内容。

1.15.3.3 模拟中继线通道的状态转移



状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	描述
Standby	0	S_CALL_STANDBY	“空闲”状态。 在通道迁移到“空闲”状态前，驱动程序会： ➢ 清空DTMF检测器的所有缓冲区 ➢ 清空 CallerID 缓冲区 ➢ 关闭DTMF发生器 ➢ 关闭信号音发生器 ➢ 清零并启动铃流信号检测器 ➢ 清零并关闭信号音检测器 ➢ 开启(如果 Caller ID 检测器工作在 DTMF 模式)或关闭(如果 Caller ID 检测器工作在 FSK 模式)DTMF 检测器 ➢ 关闭极性反转检测器(如果硬件支持极性反转信号检测功能) ➢ 将 AutoDial 任务的进展值设置为 DIAL_STANDBY ➢ 终止 WaitDtmf 任务（如果启动）
OffHook	1	S_CALL_PICKUPED	“摘机”状态。 在通道迁移到“摘机”状态前，驱动程序会： ➢ 清零并启动信号音检测器 ➢ 清零并启动铃流信号检测器 ➢ 如果 Caller ID 检测器工作在 FSK 模式，启动 DTMF 检测器
Ringing	2	S_CALL_RINGING	“振铃”状态。
Connected	3	S_CALL_TALKING	“连接”状态，呼叫双方可以进行通话。 在通道迁移到“连接”状态前，驱动程序会： ➢ 清零并启动信号音检测器 ➢ 如果 Caller ID 检测器工作在 FSK 模式，启动 DTMF 检测器 ➢ 清零铃流信号检测器的计数器；
WaitDialtone	4	S_CALL_ANALOG_WAITDIALTONE	“等待拨号音”状态。 在通道进入“等待拨号音”状态前，驱动程序会： ➢ 清零信号音检测器 ➢ 将 AutoDial 任务进展值设置为 DIAL_DIALING

Dialing	5	S_CALL_ANALOG_TXPHONUM	“拨号”状态。 在通道进入“拨号”状态前，驱动程序会： ➢ 清零并关闭信号音检测器 ➢ 关闭 DTMF 检测器
WaitRingBackTone	6	S_CALL_ANALOG_WAITDIALRESULT	“等待回铃音”状态。 在通道进入“等待回铃音”状态前，驱动程序会： ➢ 清零并启动信号音检测器 ➢ 启动增强的远端摘机检测器，详细信息请参见本章中“增强的远端摘机检测器”部分内容。 ➢ 启动 DTMF 检测器 ➢ 启动远端摘机检测器
Pending	7	S_CALL_PENDING	“未决”状态。
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP	“等待被叫用户摘机”状态。此时，被叫方应该听到振铃信号音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号。 在通道进入“等待被叫用户摘机”状态前，驱动程序会： ➢ 将 AutoDial 任务进展值设置为 DIAL_ECHOTONE，并向应用程序抛出 E_PROC_AutoDial 事件

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

驱动程序内部事件的描述如下表所示：

事件名称	描述
RingOn	驱动程序在模拟中继线上检测到了振铃信号时触发此内部事件。详细信息请参见本章中“模拟电话线上的铃流信号”部分内容。
RingOff	驱动程序在模拟中继线上检测到了振铃信号消失时触发此内部事件。详细信息请参见本章中“模拟电话线上的铃流信号”部分内容。
T1Out	T1 定时器溢出。T1 定时器由配置项MaxWaitDialToneTime进行设置，缺省值为 3 秒。 T1 溢出后，驱动程序的操作为： ➢ 设置 AutoDial 任务的进展值为 DIAL_NO_DIALTONE，并向应用程序抛出 E_PROC_AutoDial 事件； ➢ 将 PendingReason 设置为 ANALOGOUT_NO_DIALTONE，然后迁移到 Pending 状态。
T2Out	T2 定时器溢出。T2 定时器由配置项MaxWaitAutoDialAnswerTime进行设置，缺省值为 25 秒。 T2 溢出后，驱动程序的操作为： ➢ 设置 AutoDial 任务的进展值为 DIAL_NOANSWER，并向应用程序抛出 E_PROC_AutoDial 事件； ➢ 将 PendingReason 设置为 ANALOGOUT_NOANSWER，然后迁移到 Pending 状态。
Silence1	远端摘机检测器输出E_CHG_ToneAnalyze事件（携带CHKTONE_NOVOICE参数）时触发此内部事件。有关远端摘机检测器的详细信息请参见本章中“远端摘机检测器”部分内容。 驱动程序会： ➢ 将AutoDial任务进展值设置为DIAL_NOVOICE，并向应用程序抛出E_PROC_AutoDial事件 ➢ 将 PendingReason 设置为 ANALOGOUT_NOVOICE，然后迁移到 Pending 状态
Silence2	远端摘机检测器输出E_CHG_ToneAnalyze事件（携带CHKTONE_ECHO_NOVOICE参数）时触发此内部事件。有关远端摘机检测器的详细信息请参见本章中“远端摘机检测器”部分内容。 驱动程序会： ➢ 将 AutoDial 任务进展值设置为 DIAL_ECHO_NOVOICE，并向应用程序抛出 E_PROC_AutoDial 事件 ➢ 将 PendingReason 设置为 ANALOGOUT_ECHO_NOVOICE，然后迁移到 Pending 状态
DialToneDetected	信号音检测器在线路上检测到拨号音时触发此内部事件。详细信息请参见本章中“信号音检测器”部分内容。
EndofDTMF	驱动程序将 DTMF 发送缓冲区中的全部字符发送完（拨号结束）时触发此内部事件
RingBackTone	信号音检测器在线路上检测到回铃音时触发此内部事件。详细信息请参见本章中“信号音检测器”部分内容。
Answered	信号音检测器在线路上检测到下列结果时触发此内部事件： ✧ CHKTONE_VOICEF1：设置 AutoDial 任务进展值为 DIAL_VOICEF1，并向应用程序抛出 E_PROC_AutoDial 事件 ✧ DIAL_VOICEF2：设置AutoDial任务进展值为DIAL_VOICEF2，并向应用程序抛出E_PROC_AutoDial事件 ✧ DIAL_VOICE：设置 AutoDial 任务进展值为 DIAL_VOICE，并向应用程序抛出 E_PROC_AutoDial 事件

PolarReverse	如果电压检测器在线路上检测到了极性反转信号，并且配置项 DisablePolarReverse 设置为 0 时触发此内部事件。 如果通道处于 Dialing、WaitRingBackTone 或 WaitAnswer 状态时，驱动程序会将 AutoDial 任务进展值设置为 DIAL_VOICE，并向应用程序抛出 E_PROC_AutoDial 事件
BusyTone	信号音检测器在线路上检测到忙音时触发此内部事件。有关设置忙音检测参数详细信息请参见本章中“信号音检测器”部分内容。 检测到忙音时，如果通道处于 Talking 状态，驱动程序会： ➢ 将 PendingReason 设置为 ANALOGOUT_TALKING_REMOTE_HANGUPED，然后迁移到 Pending 状态。 如果是 WaitRingBackTone 或 WaitAnswer 状态，驱动程序会： ➢ 将 AutoDial 任务进展值设置为 DIAL_BUSYTONE，并向应用程序抛出 E_PROC_AutoDial 事件 ➢ 将 PendingReason 设置为 ANALOGOUT_BUSYTONE，然后迁移到 Pending 状态。

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_PROC_AutoDial	应用程序调用SsmAutoDial函数即启动了AutoDial任务。当AutoDial任务有进展时，驱动程序向应用程序抛出本事件。有关驱动程序抛出本事件的详细说明请参见前面章节

注：事件的声明在文件 ShpA3Api.h 中。

1.15.3.4 远端摘机检测器(Remote Pickup Detector)（SHT 系列 only）

远端摘机检测器只适用于模拟中继线通道，用于去话呼叫时精确检测被叫用户是否摘机。

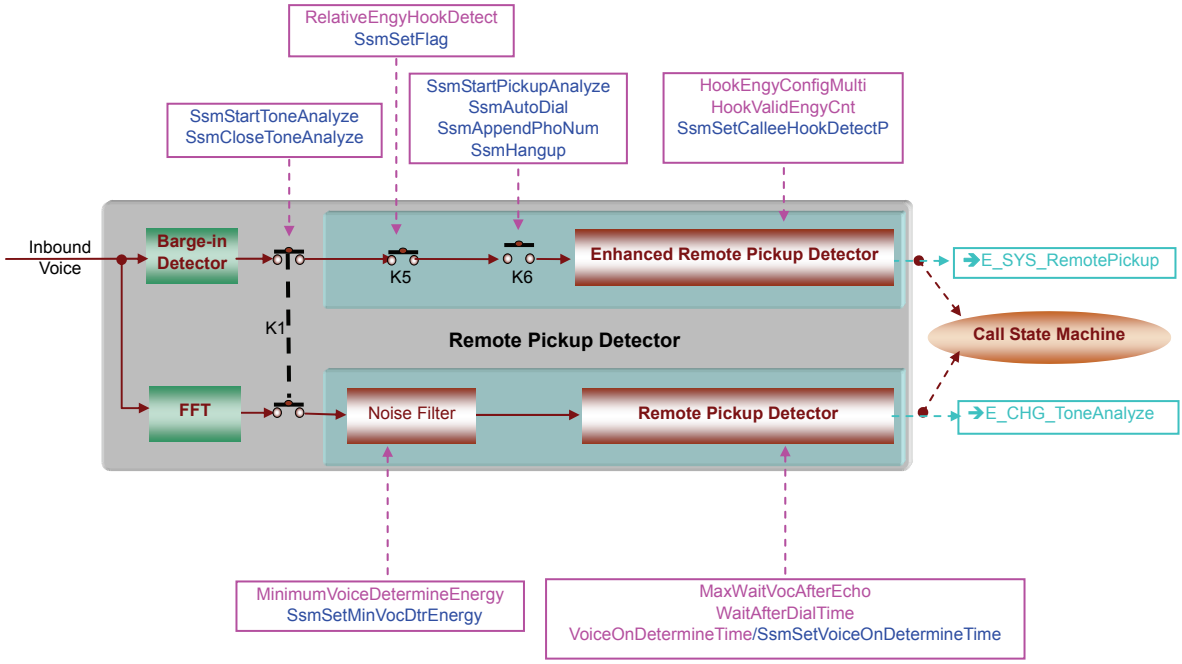
对于模拟中继线路，在进行去话呼叫时，当完成了被叫电话号码的发送后，如何判断被叫方是否摘机有时是一个比较困难的事情。目前，判断被叫摘机有 2 种方式：基于极性反转检测和基于语音活动检测的判断。

有关检测极性反转信号的详细内容请参见本章中“模拟电话线的电压变化”部分内容。

对于基于语音活动检测的远端摘机判断模式，拨号结束后大致会出现下列 2 种情形：

- 被叫方听到振铃信号后（不论时间长短），摘机，并且立即说话（比如，说“hello”）。在这种情况下，主叫通道对线路上的语音能量进行检测，当线路上的语音能量强度和持续时间大于预设门限时就认为被叫摘机。需要注意的是，当被叫方设置了彩铃时，由于交换机会用播放的彩铃信号（语音信号）作为回铃音回送给主叫方，因此这种检测方法会失效。
- 被叫方听到振铃信号后摘机，但保持静默（不说话）。在这种情况下，主叫方的线路上不会出现回铃音，也不会有任何语音信号，主叫通道通常无法判断被叫是否摘机，需要由应用程序自行进行判断。

三汇驱动程序为每个模拟中继通道都提供了独立的基于语音活动检测的远端摘机检测器，它的内部原理框图、相关的配置项和函数、输出的事件如下图所示：



远端摘机检测器包括常规的远端摘机检测器（包括上图中的Noise Filter、Frequency Detector、Remote Pickup Detector）和增强的远端摘机检测器（包括上图中的K5 开关、K6 开关和Enhanced Remote Pickup Detector）组成，由k1 开关进行控制。开关K1 可以通过函数SsmStartToneAnalyze / SsmCloseToneAnalyze来设置。**注意：开关K1 同时控制了信号音检测器和能量检测器。**

常规的远端摘机检测器使用板载 DSP 的 FFT 模块输出数据作为输入，适用于普通的应用场合。它抛出的是 E_CHG_ToneAnalyze 事件。

增强的远端摘机检测器使用一种特殊的算法来判断模拟电话线上的摘机动作，对远端摘机判断的准确度要高于常规的远端摘机检测器。它抛出的是 E_SYS_RemotePickup 事件。

1.15.3.4.1 普通的远端摘机检测器

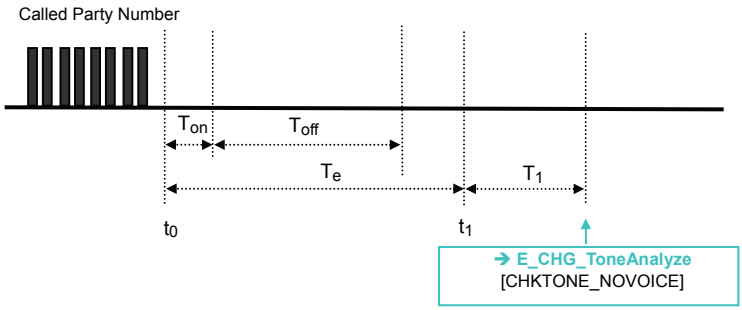
普通的远端摘机检测器检测线路上的语音活动。

噪声滤波器过滤掉线路上的噪声。它的工作原理和工作参数设置与“信号音检测器”节的噪声滤波器完全一样，详细内容请参见“信号音检测器”节的相关部分。

当噪声滤波器判定线路上存在语音信号后，只有当该语音信号不是信号音分析器中第 1 呼叫进程音检测器所指定的信号音，远端摘机检测器才认为线路上真正出现了语音。

当满足下列 3 种情况之一时远端摘机检测器就输出 E_CHG_ToneAnalyze 事件：

(1) 被叫方一听到振铃信号就摘机，然后保持静默。在这种情况下，主叫方的线路上不会出现回铃音(也可能出现短暂的回铃音信号，但因不是一个完整的周期而被过滤掉，无法检测出来)，也不会有任何语音信号，如下图所示：



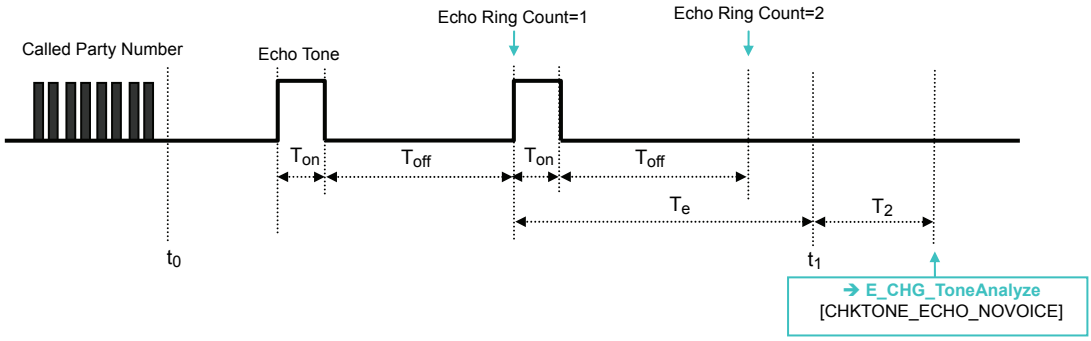
上图中，

T_{on}、T_{off}：分别表示回铃音波形参数设置的高电平持续时间和低电平持续时间；

T_e：回铃音周期的最大允许误差，T_e= (T_{on}+T_{off})*(1+20%)

T_1 ：拨号完成后的静默持续时间判定门限。从t_1 处开始，如果线路上保持静默的持续时间大于T_1，驱动程序就输出 E_CHG_ToneAnalyze（携带 CHKTONE_NOVOICE 参数）。T_1 由配置项 WaitAfterDialTime指定。

(2) 被叫方听完至少一声振铃信号后才摘机，但随后保持静默。在这种情况下，主叫通道上会检测到至少 1 次回铃音，但同样检测不到任何语音活动信号，如下图所示：



上图中，

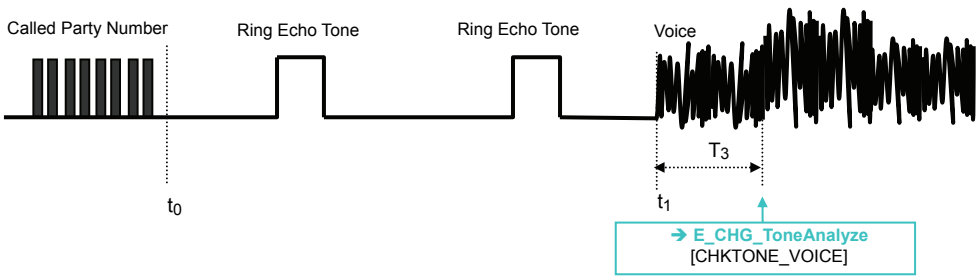
t_0：表示拨号完毕的时刻

T_{on}、T_{off}：分别表示回铃音波形参数设置的高电平持续时间和低电平持续时间

T_e：回铃音周期的最大允许误差，T_e = (T_{on}+T_{off})*(1+20%)

T_2 ：当回铃音消失后的静默持续时间判定门限。从t_1 处（即判断回铃音信号消失的时间点）开始，如果线路上保持静默的持续时间大于T_2，驱动程序就输出E_CHG_ToneAnalyze（携带CHKTONE_ECHO_NOVOICE参数）。T_2 由配置项MaxWaitVocAfterEcho 指定。

(3) 被叫方听到振铃信号后（不论时间长短），摘机，并且立即说话（比如，说“hello”）。



上图中，T_3 为语音信号判定门限。从出现语音信号的t_1 时刻开始，当语音信号持续时间大于T_3 所指定的判定门限时，驱动程序就输出E_CHG_ToneAnalyze（携带CHKTONE_VOICE参数）。T_3 由配置项VoiceOnDetermineTime或函数SsmSetVoiceOnDetermineTime指定。函数SsmGetVoiceOnDetermineTime可以用来获取驱动程序保存的T_3 的值。

1.15.3.4.2增强的远端摘机检测器

K5 开关由配置项RelativeEngyHookDetect或函数SsmSetFlag（携带F_RELATIVEENGYHOOKDETECT参数）进行设置，函数SsmGetFlag（携带F_RELATIVEENGYHOOKDETECT参数）可以用来获取K5 开关的状态。系统初始化成功后，K5 开关处于闭合状态。

K6 开关可以通过应用程序控制，也可由驱动程序自动控制。系统初始化成功后，K6 开关处于断开状态。

当满足下列条件时，K6 开关将被自动闭合以启动远端摘机检测器：

- 应用程序调用SsmStartPickupAnalyze函数；
- 应用程序调用SsmAutoDial函数发起去话呼叫后，在S_CALL_ANALOG_TXPHONUM状态，驱动程序会将被叫号码逐个发送给对端交换机。当完成全部被叫号码的发送后，通道会迁移到S_CALL_ANALOG_WAITDIALRESULT状态。在状态迁移之前，驱动程序会自动闭合K6；
- 如果通道处于S_CALL_ANALOG_TXPHONUM或S_CALL_ANALOG_WAITDIALRESULT状态时，应用程序调用了SsmAppendPhoNum函数

当满足下列条件时，驱动程序自动断开 K6 开关（启动远端摘机检测器）：

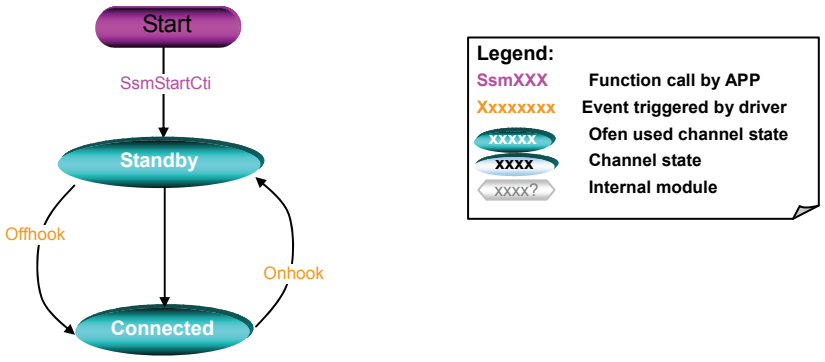
- 当远端摘机检测器完成一次检测并输出 E_SYS_RemotePickup 事件后；
- 应用程序调用SsmHangup函数；
- 模拟中继线通道的状态机收到信号音检测器抛出的“检测到忙音”事件后

配置项HookEngyConfigMulti、HookValidEngyCnt或函数SsmSetCalleeHookDetectP用于设置远端摘机检测器的工作参数。

当远端摘机检测器检测到被叫摘机后，会抛出E_SYS_RemotePickup事件，并将此事件送到呼叫状态机进行后续处理。SsmGetPickup函数可以用来取得检测结果。

1.15.4 坐席通道

1.15.4.1 状态迁移图



状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	状态描述
Standby	0	S_CALL_STANDBY	空闲状态。 在通道迁移到“空闲”状态时，驱动程序会： ➢ 清空DTMF检测器的所有缓冲区 ➢ 关闭DTMF发生器，并终止尚未完成的WaitDtmf任务 ➢ 关闭信号音发生器
OffHook	1	S_CALL_PICKUPED	“摘机”状态。 在通道迁移到“摘机”状态时，驱动程序会： ➢ 如果配置项AutoSendDialTone设置为 1，开始向话机发送拨号音 ➢ 清空DTMF检测器的所有缓冲区，并将其启动

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

注意：当通道的状态发生迁移时，驱动程序都会向应用程序抛出 E_CHG_ChState 事件。

驱动程序自动触发的内部事件如下表示：

事件	事件描述
Offhook	当驱动程序检测到话机的摘机动作时触发此内部事件
Onhook	当驱动程序检测到话机的挂机动作时触发此内部事件

1.15.4.2 检测话机的摘机/挂机动作

当话机上发生摘机或挂机动作时，线路上可能会产生一些抖动信号。为了防止误检测，需要对其进行滤波处理，以消除抖动信号对摘机、挂机检测的准确性。配置项UserOnHookFilterTime用于设置摘机信号的最小保持时间，只有当驱动程序检测到摘机信号的保持时间大于该配置项的设定值时，驱动程序才会判定为摘机。

每当驱动程序判定摘机动作或挂机动作时，都会向应用程序抛出E_CHG_HookState事件。函数SsmGetHookState也可以用来获取坐席通道的摘/挂机状态。

1.15.4.3 检测话机的闪断信号

由于坐席通道用于模拟交换机上用户模块电路的特性，因此Synway板卡的坐席通道具有检测电话机上产生的闪断信号的能力。有关闪断信号的详细介绍请参见本章中“模拟电话线上的闪断信号”部分内容。

函数SsmSetLocalFlashTime 或配置项MaxLocalFlashTime用于设置判定闪断信号的最大保持时间N。当用户按下话机的叉簧，在保持一定时间（假定为n）后再次放开叉簧，如果n<N，驱动程序判定用户发送了一个闪断信号，否则认为是用户的挂机动作。

驱动程序内部为每个坐席通道配备了闪断信号计数器，每当检测到一次闪断信号时，该计数器就加 1，同时向应用程序抛出E_CHG_FlashCount事件。函数SsmGetFlashCount也可以用来获取闪断信号的计数器值。每当坐席通道的状态迁移到“通话”或“挂机”时，驱动程序都会将计数器自动清零。函数SsmClearFlashCount也可用于将计数器清零。

1.15.4.4 向电话机发送振铃信号

Synway板卡的坐席通道都具有铃流发生器，用于向电话机发送振铃信号。常规振铃信号的波形特征为 1 秒高电平、4 秒低电平的周期性信号，但配置项UserChGenerateRingMode可以将振铃信号的波形设置为持续性的信号。

SsmStartRing、SsmStartRingWithCIDStr和SsmStartRingWithFskCID均可以用来向话机发送振铃信号。SsmStartRing可以向模拟电话机和磁石话机发送振铃信号，SsmStartRingWithCIDStr和SsmStartRingWithFskCID还可以向话机发送主叫号码信息，但只用于坐席通道。SsmStartRingWithCIDStr的输入为已经调制好的FSK比特流，因此通常与函数fPcm_ConvertFskCID一起使用，先调用fPcm_ConvertFskCID将字符串转换成FSK比特流，再调用

SsmStartRingWithFskCID进行发送。

驱动程序为每个坐席通道或磁石话机通道设置了一个发送振铃信号的计数器，每当完成一次振铃信号的发送（高电平），会将计数器加 1。函数SsmCheckSendRing可以用来

获取该计数器的值。

在任何时刻，应用程序都可以调用函数SsmStopRing来停止向话机发送振铃信号。

1.15.4.5 在电话线上产生极性反转信号

有关极性反转信号的详细内容请参见本章中“模拟电话线的电压变化”部分。

Synway 板卡的坐席通道是否具有在模拟电话线上产生极性反转信号的功能，取决于下面 2 个因素：

- 坐席模块所在的板卡的馈电电压为-48V
- 坐席模块所在的板卡为下列型号之一：
  - SHT-8B/PCI （**1.0** 版）
  - SHT-8B/PCI/FAX （**1.0** 版）
  - SHT-16B-CT/PCI （**1.0** 版）
  - SHT-16B-CT/PCI/FAX （**1.0** 版）
  - SHT-16B-CT/PCI/MP3 （**1.0** 版）
  - SHT-16B-CT/cPCI （**1.0** 版）
  - SHT-16B-CT/cPCI/MP3 （**1.0** 版）
  - SHT-16B-CT/cPCI/FAX （**1.0** 版）

通道要使用极性反转信号的功能，需要通过配置项UserSendPolar进行设置。

函数SsmGetPolarState用于获取电话线上的当前极性，函数SsmSetPolarState用于设置电话线上的电压极性。

1.15.4.6 检测到话机摘机后自动送拨号音

当与坐席通道相连的电话机摘机后，是否由驱动程序自动向电话机发送拨号音，由配置项AutoSendDialTone决定。在发送拨号音期间，如果检测到DTMF字符，是否由驱动程序自动停止拨号音的发送，由配置项StopSendDialToneOnDtmf决定。

1.15.5 联合模块

对于 SHT 系列板卡，Synway 公司还提供一种特殊的业务模块的组合，即将一个模拟中继线通道和一个坐席通道的接入电路安装到一个业务模块上，称为联合模块。联合模块最大的特点，就是具有“停电直通”功能，当应用系统因故停电时，联合模块上的模拟中继线通道与坐席通道是直通的，因此，坐席通道仍然能够经由模拟中继线通道与电话网进行通讯。

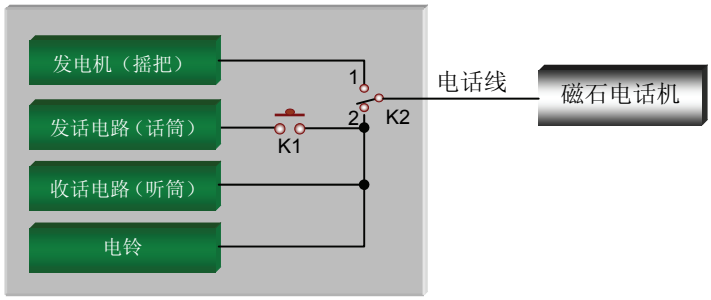
函数SsmSetUnimoduleState或配置项UnimoduleState用于设置联合模块上模拟中继线通道和坐席通道之间的连接关系。

1.15.6 磁石电话机通道

1.15.6.1 概述

在 SHT 系列板卡上安装了磁石模块的通道称为磁石电话机通道，简称磁石通道，用于连接磁石电话机。

磁石模块的内部结构与一个磁石电话机相似。磁石电话机的工作原理如下图所示：



磁石电话机内部由发话电话、收话电路、电铃、发电机和 K1、K2 两个开关组成（图中各部件均以单线方式示意连接，实际为双线构成回路）。当摇动发电机手柄，K2 自动跳至 1 点，将铃流送至对方话机的电铃。按下握柄 K1（相当于普通话机的叉簧松开），发话电路接通，可以发话（设置 K1 的主要作用是不发话时断开干电池）。

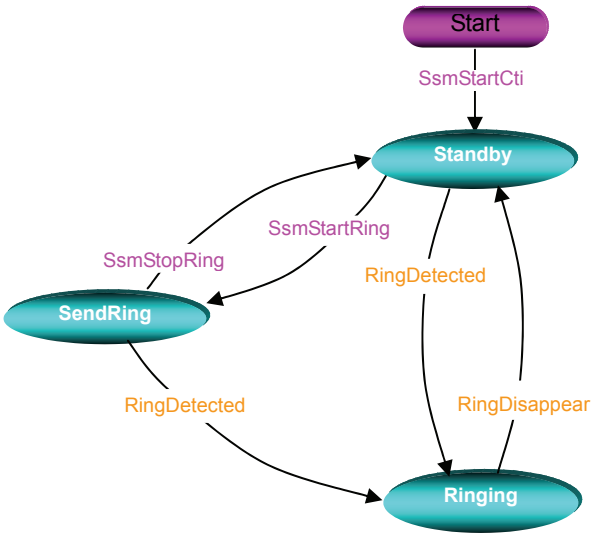
磁石模块的特点：

- 磁石模块只有空闲、发送振铃和振铃三个状态，不能执行摘机、挂机操作。发送铃流信号的节奏与坐席模块相同，即响 1 秒、停 4 秒，在 4 秒钟的低电平期间仍可以正常检测振铃；
- 内置铃流信号发生器，可以在线路上产生铃流信号，其工作原理与操作方法与 SHT 系列 B 型板卡及以后版本的坐席模块的铃流发生器相同。如果在线路上已经存在着铃流信号，则不能在磁石模块上发送铃流信号；如果磁石模块在发送铃流信号时在线路上也检测到了铃流信号，则磁石模块会自动停止本端发送的铃流信号；
- 磁石模块省略了 K1，即当处于空闲和振铃状态时，收发话电路总是接通的；

注意：

由于板卡上的业务模块识别电路不能识别磁石模块，而是会将其识别为模拟中继线业务模块，因此，要使用磁石模块，必须对配置项IsMagnetModule进行设置，以便驱动程序能够正确区分这 2 种业务模块。

1.15.6.2 状态迁移图



上表中，

状态标识	值	宏定义	状态描述
Standby	0	S_CALL_STANDBY	空闲状态。驱动程序初始化成功后，进入本状态。
SendRing	100	S_CALL_SENDRING	发送振铃状态。通道正在向磁石电话发送振铃。同时检测是否有

			铃流。铃流的检测可以通过配置项RingDetectFilterPara进行设置。
Ringing	2	S_CALL_RINGING	振铃状态。磁石电话正在振铃。驱动检测振铃是否消失。

1.15.7 使用未安装业务模块的通道

SHT系列板卡采用模块化结构，由底板和具有不同功能的业务模块组成。如果在板卡上没有安装任何业务模块，底板上的DSP芯片仍然能够正常工作，因此，仍然能够提供一定的语音处理功能。但是，由于没有电话线的物理接口电路，因此，它的语音信号只能来自TDM交换总线。在某些特殊的应用场合，当其它具有TDM交换总线的板卡需要录音功能时，可把语音数据通过TDM总线交换到到未安装业务模块的通道，将Barge in检测器的输出作为录音操作的控制条件，即可进行录音操作。配置项NoModuleChBusRec或函数SsmSetNoModuleChBusRec用于提供这种功能。未安装业务模块通道具有的语音处理能力包括：录音、TDM总线交换、DTMF接收及BargeIn检测功能。

注意：当 SHT 系列板卡的底板上没有安装任何业务模块时，通道的类型编号为 20，表示无模块。

1.16 SHD 系列板卡（CTI 系列）

1.16.1 SHD 系列板卡简介

SHD 系列板卡支持的功能如下表所示：

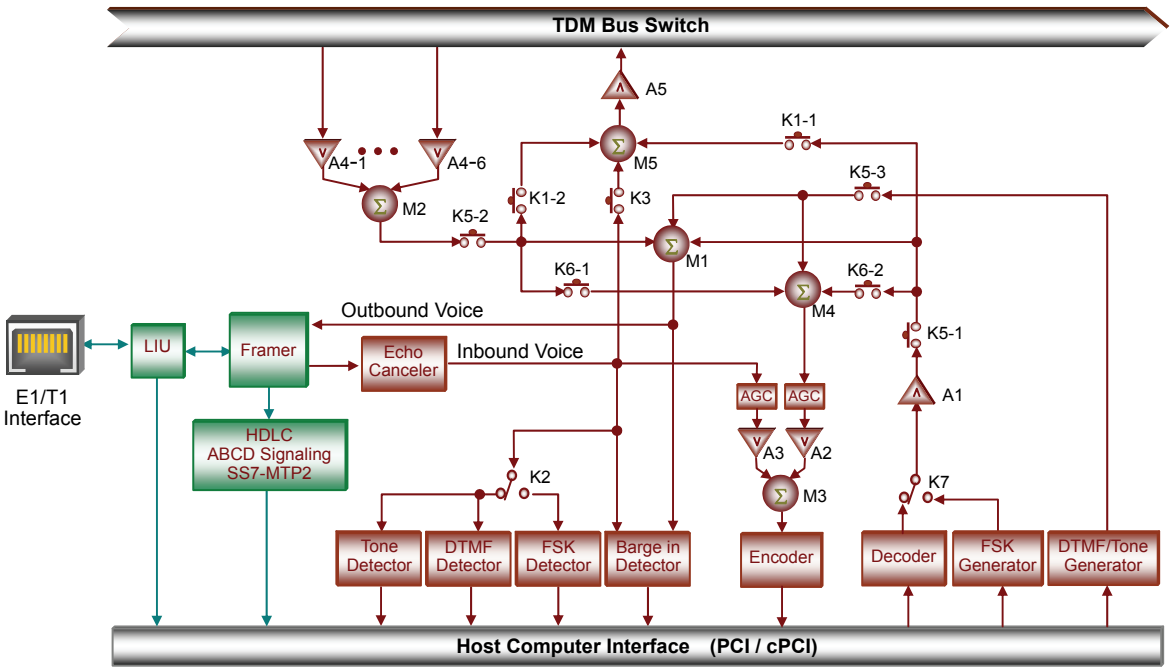
Board Model	Form Factor	Trunk Type	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	FSK	Max Fax Resource	SS1	ISDN PRI	SS7	SS7 Signaling Links
SHD-30A-CT/PCI/SS1	PCI	1 E1	30	√	√	√	N/A	√	N/A	N/A	N/A
SHD-30A-CT/PCI/ISDN	PCI	1 E1	30	√	√	√	N/A	√	√	N/A	N/A
SHD-30A-CT/PCI/SS7	PCI	1 E1	30	√	√	√	N/A	√	√	√	1
SHD-30B-CT/PCI/SS7/FAX	PCI	1 E1	30	√	√	√	24	√	√	√	1
SHD-60A-CT/PCI/SS1	PCI	2 E1	60	√	√	√	N/A	√	N/A	N/A	N/A
SHD-60A-CT/PCI/ISDN	PCI	2 E1	60	√	√	√	N/A	√	√	N/A	N/A
SHD-60A-CT/PCI/SS7	PCI	2 E1	60	√	√	√	N/A	√	√	√	2
SHD-60B-CT/PCI/SS7/FAX	PCI	2 E1	60	√	√	√	16	√	√	√	2
SHD-120A-CT/PCI/SS1	PCI	4 E1	120	√	√	√	N/A	√	N/A	N/A	N/A
SHD-120A-CT/PCI/ISDN	PCI	4 E1	120	√	√	√	N/A	√	√	N/A	N/A
SHD-120A-CT/PCI/SS7	PCI	4 E1	120	√	√	√	N/A	√	√	√	4
SHD-120D-CT/PCI	PCI	4 E1/T1/J1	120/92	√	√	√	N/A	N/A	√	√	4
SHD-120D-CT/PCI/EC	PCI	4 E1/T1/J1	120/92	√	√	√	N/A	N/A	√	√	4
SHD-240D-CT/PCI	PCI	8 E1/T1/J1	240/184	√	√	√	N/A	N/A	√	√	8
SHD-240D-CT/PCI/EC	PCI	8 E1/T1/J1	240/184	√	√	√	N/A	N/A	√	√	8
SHD-30C-CT/PCI	PCI	1 E1/T1	30/23	√	√	√	N/A	√(E1Only)	√	√(E1Only)	16
SHD-60C-CT/PCI	PCI	2 E1/T1	60/46	√	√	√	N/A	√(E1Only)	√	√(E1Only)	16
SHD-30A-CT/cPCI/SS7	cPCI	1 E1	30	√	√	√	N/A	√	√	√	1
SHD-30B-CT/cPCI/SS7/FAX	cPCI	1 E1	30	√	√	√	24	√	√	√	2
SHD-60A-CT/cPCI/SS7	cPCI	2 E1	60	√	√	√	N/A	√	√	√	2
SHD-60B-CT/cPCI/SS7/FAX	cPCI	2 E1	60	√	√	√	16	√	√	√	4
SHD-120A-CT/cPCI/SS7	cPCI	4 E1	120	√	√	√	N/A	√	√	√	4
SHD-240A-CT/cPCI/SS7	cPCI	8 E1	240	√	√	N/A	N/A	N/A	√	√	8
SHD-240S-CT/cPCI/SS7	cPCI	8 E1	240	√	N/A	N/A	N/A	N/A	√	√	8

SHD-480A-CT/cPCI/SS7	cPCI	16 E1	480	√	√	N/A	N/A	N/A	√	√	16
SHD-480S-CT/cPCI/SS7	cPCI	16 E1	480	√	N/A	N/A	N/A	N/A	√	√	16
SHD-240C-CT/cPCI	PCI	8 E1/T1	240/184	√	√	√	N/A	N/A	√	√	8
SHD-240C-CT/cPCI/EC	PCI	8 E1/T1	240/184	√	√	√	N/A	N/A	√	√	8
SHD-480C-CT/cPCI	PCI	16 E1/T1	480/368	√	√	√	N/A	N/A	√	√	16
SHD-480C-CT/cPCI/EC	PCI	16 E1/T1	480/368	√	√	√	N/A	N/A	√	√	16

注：SHD 系列板卡均具有 TDM 交换总线功能，因此未在上图中列出。

1.16.2 SHD 系列板卡的工作原理框图

SHD 系列板卡的工作原理图如下图所示：



原理框图中有关录音操作的符号说明：

名称	描述
M1	去话混音器
K5-3 [1]	信号音发生器、DTMF发生器的信号输出控制开关，由驱动程序自动控制，通常处于断开状态，只有当应用程序调用SsmTxDtmf函数启动DTMF发生器，或者调用SsmSendTone/ SsmSendToneEx函数启动信号音发生器后，驱动程序才会自动闭合K5-3 开关；上述任务结束后，K5-3 自动断开。
M2	下总线信号混音器
A4-1 … A4-6	下总线信号进入M2 前的音量调节器，可以通过函数SsmSetListenVlmInConf进行设置，缺省值为 0DB。
K5-2 [1]	下总线信号混音器的输出信号控制开关，由驱动程序自动控制。当应用程序调用的总线连接函数中包含下总线的操作时，驱动程序会自动闭合K5-2 开关；当M2 的输入全部被断开时，驱动程序自动断开K5-2。有关交换总线的函数请参见本章中交换总线部分内容。
M3	录音混音器。 M3 的输出进入编码器（Encoder）模块，然后形成录音数据。
A3	来话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecVolume进行设置，也可以通过配置项DefaultRecordVolume设置，缺省值为 0。
A2	去话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecMixer进行设置，也可以通过

	配置项DefaultRecordMixerVolume设置，缺省值-7，关闭A2。
M4	录音用途的去话混音器。 用于将下总线信号混音器的输出、放音数据、信号音发生器/DTMF 发生器的输出进行混音，混音结果进入录音混音器 M3。
K6-1 K6-2	开关K6-1 控制总线混音器M2 的输出是否作为录音混音器M3 的一个输入信号源；开关K6-2 控制放音数据是否作为录音混音器M3 的一个输入信号源。函数SsmSetRecBack用来设置这 2 个开关。
M5	上总线混音器。 用于将放音数据、下总线信号混音器 M2 的输出、来话信号进行混音，混音结果上相到 TDM 交换总线。板卡初始化时，会将M5 的输出上行到TDM总线的某个时隙（记为ts）上，函数SsmLinkToBus可以用来改变ts。
A5	M5 进入总线前的音量增益，可以通过配置项DefaultSpeakVolume或下列函数进行设置： ✧ SsmTalkWithEx ✧ SsmListenToEx / SsmLinkFromEx / SsmLinkFromAllCh
K3	来话信号是否进入上总线混音器M5 的控制开关，可通过函数SsmSetFlag（携带F_InVoiceToBus参数）或配置项InVoiceToBus进行设置，缺省值为闭合。
K1-1 [1]	控制放音数据是否进入混音器M5，进而上行到TDM交换总线上的开关，通常为断开状态，可以通过函数SsmSetPlayDest进行控制。K1-1 只能用于电话会议中给会议室播放背景音的用途。
K1-2	M2 的输出信号是否重新回到TDM交换总线的控制开关，可以通过函数SsmSetFlag（携带F_MixerResToBus参数）进行设置，缺省值为断开
	与放音操作相关的模块
K7	语音播放/FSK数据发送器的切换开关，由驱动程序自动控制。当应用程序调用播放函数时，K7 指向语音编码器模块Decoder；当应用程序调用SsmStartSendFSK函数启动发送FSK数据时，K7 执行FSK Generator。有关语音播放的函数请参见语音播放部分内容。
A1	放音的音量调节器，可以通过配置项DefaultPlayVolume设置，也可通过函数SsmSetPlayVolume或SsmSetPlayGain进行调节，缺省的音量设置为 0。
K5-1 [1]	控制放音操作的开关，由驱动程序自动控制。当应用程序调用播放函数时，驱动程序自动开启K5-1 开关；当放音任务结束时，驱动程序自动关闭K5-1。有关语音播放的函数请参见语音播放部分内容。
K2	FSK 检测器与其它检测器之间的切换开关，通常指向其它的检测器，只有在开启 FSK 检测器期间才会指向 FSK 检测器。

注[1]：

- **对于SHD系列的某些型号的板卡，由于板载DSP芯片的资源限制，当K1-1 断开时，K5-1、K5-2、K5-3 中只能有一个闭合。因此，当通道正在进行放音操作（此时K5-3 已经闭合）或启动了信号音发生器（此时K5-1 已经闭合）时，如果应用程序调用了会引发K5-2 闭合的总线操作函数，如果配置项LinkFromStopPlayAndTone设置为 1，放音操作将会被立即终止，K5-2 立即闭合；如果配置项LinkFromStopPlayAndTone设置为 0，K5-2 在放音操作正常终止后才会闭合。引发K5-2 闭合的总线操作函数包括：**

  - SsmTalkWith / SsmTalkWithEx
  - SsmListenTo / SsmListenToEx / SsmLinkFrom / SsmLinkFromEx / SsmLinkFromAllCh
  - SsmLinkFromBus / SsmLinkFromBusEx

  涉及的型号包括：

  - SHD-30A-CT/PCI/SS1
  - SHD-30A-CT/PCI/ISDN
  - SHD-30A-CT/PCI/SS7
  - SHD-30B-CT/PCI/SS7/FAX

- SHD-60A-CT/PCI/SS1
- SHD-60A-CT/PCI/ISDN
- SHD-60A-CT/PCI/SS7
- SHD-60B-CT/PCI/SS7/FAX
- SHD-120A-CT/PCI/SS1
- SHD-120A-CT/PCI/ISDN
- SHD-120A-CT/PCI/SS7
- SHD-30A-CT/cPCI/SS7
- SHD-30B-CT/cPCI/SS7/FAX
- SHD-60A-CT/cPCI/SS7
- SHD-60B-CT/cPCI/SS7/FAX
- SHD-120A-CT/cPCI/SS7

- **如果 K1-1 闭合，则 K5-1、K5-2 可以同时闭合，但 K5-3 必须保持断开。**

➢ **放音操作**

应用程序调用播放函数后， K7 指向语音解码器 Decoder， K5-1 闭合。解码后的语音数据经过音量调节器 A1 进行音量调节后，可以有 3 个去向：

✧ 进入 M1 混音器，与其它信号源混音后形成去话信号；

✧ 在 K6-2 控制下，进入 M4 混音器，与其它信号源混音后形成录音操作的信号源；

✧ 在 K1-1 控制下，进入混音器 M5，与其它信号源混音后形成上总线的信号源，只能用于电话会议中给会议室播放背景音。

➢ **录音操作的信号源选择范例**

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

✧ 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);          // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);    // 关闭 A2
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

✧ 范例 2：ch1 与 ch2 建立了双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）＋总线混音器 M2 输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);            // 建立双向连接: ch1↔ch2，K5-2 闭合
SsmSetRecBack(ch1,2);             // K6-1 闭合，K6-2 断开
SsmSetRecVolume(ch1, 2);          // 开启 A3，设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);    // 开启 A2，设置增益为-1*3DB=-3DB
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

✧ 范例 3：ch1 以主持人身份加入了一个电话会议室（假设会议室编号为 n），在 ch1 上播放背景音乐，同时通过 ch1 对电话会议进行录音（包括背景音乐）。所有信号源均为正常音量，即增益为 0。实现代码为：

```
SsmJoinConfGroup(n, ch1, ……);     // ch1 加入会议室：K5-2 闭合，K3 闭合
SsmSetPlayDest(int ch, 1);        // 闭合 K1-1
SsmSetRecBack(ch1,1);             // K6-1 闭合，K6-2 闭合
SsmPlayFile(ch1, ……);             // ch1 启动放音任务：K7 指向 Decoder，K5-1 闭合
SsmSetRecVolume(ch1, 0);          // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, TRUE, 0);     // 开启 A2，设置增益为 0DB
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

1.16.3 设置来话呼叫的收号规则

来话呼叫过程由通道的状态机自动实现，应用程序只需要预先设置好收号规则即可。收号规则是指当出现来话呼叫时，本端接收对端交换机发送的被叫方电话号码、主叫方电话号码等信息的规则。

来话呼叫的收号规则有 2 种工作模式，由配置项DefaultRcvPhoNumLen进行设置：

- ✧ 定长方式：指本端的电话号码只有一个，并且具有固定的长度。例如，“110”。当驱动程序收齐了设定长度的被叫号码时，就认为收号过程结束。如果收号规则设置为定长方式，配置项DefaultRcvCallerID和RcvPhoNumCfgLen起作用；
- ✧ 入局字冠方式：可以为本端的应用程序设置多个被叫号码，每个号码都以一个字冠开头。如果收号规则设置为入局字冠方式，配置项MaxPhoNumRule和Rule用于设置具体的收号规则。

TUP 通道、ISUP 通道、ISDN 通道和 SS1 通道均可以独立设置收号规则：

- ✧ TUP 通道的收号规则在[TUP]节中进行设置；
- ✧ ISUP 通道的收号规则在[ISUP]节中进行设置；
- ✧ ISDN 通道的收号规则在[ISDN]节中进行设置；
- ✧ SS1 通道的收号规则在[SS1]节中进行设置；

1.16.4 SS7 协议的编程

1.16.4.1 OPC 和 DPC

信令点(Signaling Point, SP)是信令网中传送各种信令消息的源点和目的点。对于国际信令点编码，CCITT 建议采用 14bit 编码，编码方式见下图：

3bit	8bit	3bit
大区或洲 （ZONE）	国家或地区 （AREA）	信令点 （POINT）

中国的信令点编码采用 24bit 方式，各比特的含义如下图所示：

8bit	8bit	8bit
主信令区编码	分信令区编码	信令点编码

Synway 的 SHD 系列板卡同时支持国际信令点编码方式和中国信令点编码方式。

信令消息的起源点称为 OPC（Originating Point Code），信令消息的目的点称为 DPC（Destination Point Code）。OPC、DPC 均在 Ss7Server.ini 文件中设置。

1.16.4.2 时隙分配

SHD 系列板卡的 PCM 时隙分配如下表所示：

Board Model	Sync. Time Slot	Voice Time Slot	Signaling Time Slot		
			SS1	ISDN PRI	SS7
SHD-30A-CT/PCI/SS1	0	1-15,17-31	16	N/A	N/A
SHD-30A-CT/PCI/ISDN	0	1-15,17-31	16	16	N/A
SHD-30A-CT/PCI/SS7	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-30B-CT/PCI/SS7/FAX	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-60A-CT/PCI/SS1	0	1-15,17-31	16	N/A	N/A

SHD-60A-CT/PCI/ISDN	0	1-15,17-31	16	16	N/A
SHD-60A-CT/PCI/SS7	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-60B-CT/PCI/SS7/FAX	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-120A-CT/PCI/SS1	0	1-15,17-31	16	N/A	N/A
SHD-120A-CT/PCI/ISDN	0	1-15,17-31	16	16	N/A
SHD-120A-CT/PCI/SS7	0	1-15,17-31	16	16	16[1]
SHD-120D-CT/PCI	0	1-15,17-31	N/A	16	16[1]
SHD-120D-CT/PCI/EC	0	1-15,17-31	N/A	16	16[1]
SHD-240D-CT/PCI	0	1-15,17-31	N/A	16	16[1]
SHD-240D-CT/PCI/EC	0	1-15,17-31	N/A	16	16[1]
SHD-30C-CT/PCI	0(E1) N/A(T1)	1-15,17-31(E1) 0-22(T1 ISDN)	16(E1)	16(E1) 23(T1)	16[1](E1)
SHD-60C-CT/PCI	0(E1) N/A(T1)	1-15,17-31(E1) 0-22(T1 ISDN)	16(E1)	16(E1) 23(T1)	16[1](E1)
SHD-30A-CT/cPCI/SS7	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-30B-CT/cPCI/SS7/FAX	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-60A-CT/cPCI/SS7	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-60B-CT/cPCI/SS7/FAX	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-120A-CT/cPCI/SS7	0	1-15,17-31	16	16	16[1]
		2-31	-	-	1[1]
SHD-240A-CT/cPCI/SS7	0	1-15,17-31	N/A	16	16[1]
SHD-240S-CT/cPCI/SS7	0	1-15,17-31	N/A	16	16[1]
SHD-480A-CT/cPCI/SS7	0	1-15,17-31	N/A	16	16[1]
SHD-480S-CT/cPCI/SS7	0	1-15,17-31	N/A	16	16[1]
SHD-240C-CT/cPCI	0	1-15,17-31	N/A	16	16[1]
SHD-240C-CT/cPCI/EC	0	1-15,17-31	N/A	16	16[1]
SHD-480C-CT/cPCI	0	1-15,17-31	N/A	16	16[1]
SHD-480C-CT/cPCI/EC	0	1-15,17-31	N/A	16	16[1]

Note [1]： 数字中继线上是否包含SS7 信令链路，以及信令链路具体由哪个时隙承载，由配置项UseTS16AsCircuit和Ss7SignalingTS决定。

1.16.4.3 设置地址信号中备用值的对应字符

在TUP和ISUP协议中，被叫号码、主叫号码等信息在消息中不是按照标准的ASCII字符进行传输，而是使用编码值。而驱动程序返回给应用程序的则是标准的ASCII字符形式的主叫号码、被叫号码，因此，需要设置一个翻译规则，将消息中的编码值翻译成统一的ASCII字符。配置项AddressSignal用于实现将地址信号中的备用值映射到特定的字符，如下表所示：

地址信号的值		对应字符	属性
16 进制	10 进制		
0	0	‘0’	不可更改
1	1	‘1’	不可更改
2	2	‘2’	不可更改
3	3	‘3’	不可更改
4	4	‘4’	不可更改
5	5	‘5’	不可更改
6	6	‘6’	不可更改

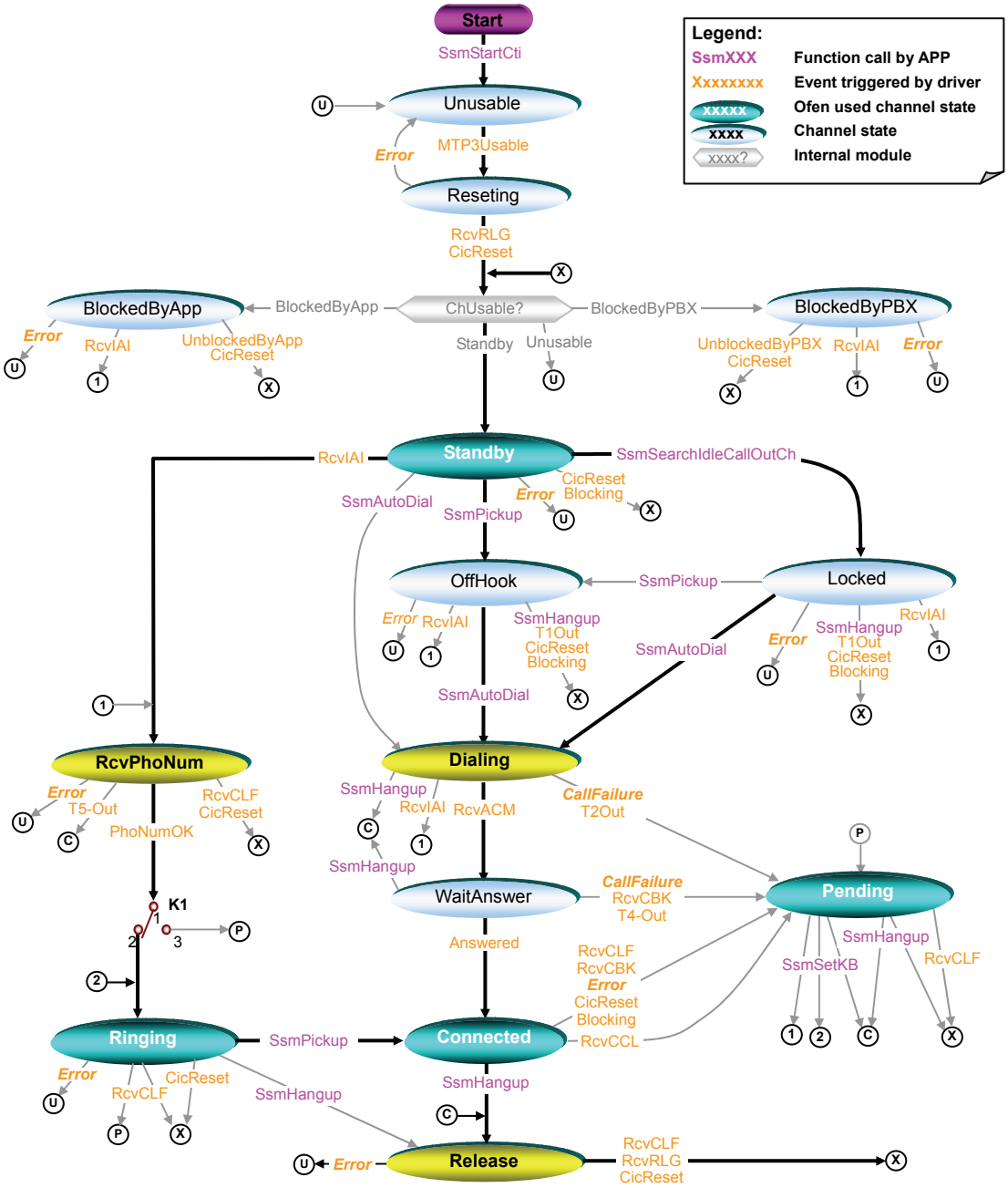
7	7	‘7’	不可更改
8	8	‘8’	不可更改
9	9	‘9’	不可更改
a	10	备用	可通过配置项AddressSignal进行设置
b	11	备用	可通过配置项AddressSignal进行设置
c	12	备用	可通过配置项AddressSignal进行设置
d	13	备用	可通过配置项AddressSignal进行设置
e	14	备用	可通过配置项AddressSignal进行设置
f	15	ST 信号	不可更改

1.16.4.4 SS7 信令应用系统的配置实例

有关SS7 信令应用系统的配置实例，请参见第 4 章中“SS7 信令服务器的应用及配置实例”部分内容。

1.16.4.5 TUP 协议的编程与应用

1.16.4.5.1 TUP 通道的状态转移



驱动程序内部事件的描述如下表所示：

事件名称	描述
Mtp3Usable	驱动程序收到 SS7 Server 的信令可用消息后，会向所有通道的状态机发送本事件，使通道从 **Unusable** 迁移到 **Reseting** 状态，开始进行电路复原。

<table>
<tr><td>CicReset</td><td>电路复原成功，当驱动程序收到电路复原消息（GRA、GRS 或 RSC 消息）时触发本事件。
驱动程序的操作：如果本通道已经将对端闭塞，或者已经向对端发送了闭塞消息并等待对端响应应答消息，则驱动程序会向对端发送 BLO 消息，以保持对对端的闭塞，同时向应用程序抛出 E_CHG_RemoteChBlock 事件。</td></tr>
<tr><td>Blocking</td><td>本通道被本端闭塞或被对端闭塞。触发条件为：
● 对端闭塞：驱动程序收到对端交换机的群闭塞消息（MGB、SGB、HGB）或电路闭塞消息（BLO）
● 本端闭塞：应用程序调用SsmBlockLocalCh或SsmBlockLocalPCM函数。</td></tr>
<tr><td>UnblockedByPBX</td><td>解除对端闭塞：驱动程序收到对端交换机的 MGU、HGU、SGU、UBL 消息时触发本事件。</td></tr>
<tr><td>UnblockedByApp</td><td>解除本端闭塞：应用程序调用解除本端闭塞函数SsmUnblockLocalPCM或SsmUnblockLocalCh后触发本事件。</td></tr>
<tr><td>RcvIAI</td><td>驱动程序收到对端交换机的来话呼叫建立请求，即 IAI 或 IAM 消息。驱动程序是否接受此来话呼叫请求，取决于通道的状态：
<table>
<tr><th>状态</th><th>描述</th></tr>
<tr><td>BlockedByPBX
BlockedByApp</td><td>在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫</td></tr>
<tr><td>Dialing</td><td>在去话呼叫过程中收到对端交换机的 IAI/IAM 消息，表明发生了“同抢”。根据 TUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则此 IAI/IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫。</td></tr>
<tr><td>OffHook
Locked</td><td>通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应</td></tr>
<tr><td>Standby</td><td>接受来话呼叫</td></tr>
<tr><td>其它</td><td>拒绝</td></tr>
</table>
如果驱动程序接受来话呼叫，在将 IAM 消息中的被叫号码信息保存到缓冲区后，向应用程序抛出 E_CHG_RxPhoNumBuf 事件，然后将通道迁移到 RcvPhoNum 状态。</td></tr>
<tr><td>RcvRLG</td><td>驱动程序收到对端交换机的 RLG 消息（释放监护信号，Release-guard signal）。</td></tr>
<tr><td>RcvCLF</td><td>驱动程序收到对端交换机的 CLF 消息。当本事件发生时，如果通道处于
● RcvPhoNum、Ringing 状态：
表明在来话呼叫的建立过程中，对端交换机（主叫方）取消了本次呼叫。
如果通道处于Ringing状态，如果配置项RingToPending设置为 1，驱动程序会设置PendingReason为 PEND_RemoteHangupOnRinging，并将通道迁移到Pending状态；否则，则驱动程序开始拆除本次来话呼叫。
● Release 状态中的 S_TUP_WaitCLF 子状态：
表明通道在拆线时收到了预期中的 CLF 消息，拆线过程完成。
● Pending 状态：
不管进入 Pending 状态的原因如何，在 Pending 状态收到 CLF 消息都将导致驱动程序自动发送 RLG 消息，并开始拆除连接。
● Connected 状态：
对于来话呼叫，在 Connected 状态收到 CLF 消息表明主叫用户先挂机。驱动程序会将 PendingReason 设置为 PEND_RemoteHangupOnTalking 后，然后将通道迁移到 Pending 状态，但不会发送 RLG 消息。</td></tr>
<tr><td>RcvACM</td><td>驱动程序收到对端交换机的 ACM 消息（被叫地址全）。在去话呼叫接续过程中，当应用程序调用 SsmAutoDial 函数启动一次去话呼叫任务后，驱动程序会向对端交换机发送 IAI、IAM 消息。如果被叫方交换机认为已经完成被叫号码等必要信息的接收，会向本端发送 ACM 消息。
驱动程序只在 Dialing、AppSetCallerID 状态响应本事件。当收到 ACM 消息后，驱动首先保存 ACM 消息，接着将 AutoDial 任务的进展设置为 DIAL_ECHOTONE，然后迁移到 WaitAnswer 状态。函数 SsmGetKB 可以用来获取收到的 ACM 的信息。</td></tr>
<tr><td>Answered</td><td>当驱动程序收到对端交换机的应答信号时触发本事件。去话呼叫时，当对端交换机处于向被叫用户振铃的状态时，如果被叫用户摘机，对端交换机会向本端发送应答信号。应答信号是指下列消息之一：
✧ ANU 消息（应答信号、计费未说明）
✧ ANC（应答信号、计费）
✧ ANN（应答信号、免费）
函数SsmGetTupFlag可以获取应答信号使用的具体消息。
本事件会导致通道从 WaitAnswer 状态迁移到 Connected 状态。在迁移到新的状态前，驱动程序还会自动进行下列操作：
✧ 如果配置项AlwaysEnableRxDtmf设置为 0， DTMF检测器将被自动启动；
✧ 如果配置项AlwaysDetectBargeIn设置为 0， Barge-in 检测器将被自动启动；
✧ 信号音检测器将被清零并自动启动
✧ 将 AutoDial 任务的进展设置为 DIAL_VOICE，任务结束，并向应用程序抛出 E_PROC_AutoDial 事件。</td></tr>
</table>

RcvCBK	驱动程序收到对端交换机的 CBK 消息。当本事件发生时，如果通道处于 ● **Connected** 状态： 表明在通话过程中检测到远端用户先挂机（去话方向）。驱动程序将 PendingReason 设置为 PEND_CalleeHangupOnTalking，然后将通道状态迁移到 Pending； ● **WaitAnswer** 状态： 表明被叫用户拒绝了本次呼叫，AutoDial 任务结束。驱动程序执行下列操作： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，将 AutoDial 任务的失败原因设置为 ATDL_RcvCBK，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将 PendingReason 设置为 PEND_CalleeHangupOnWaitRemotePickUp，然后将通道状态迁移到 Pending；
RcvCCL	驱动程序收到对端交换机的 CCL 消息，表明在 **Connected** 状态，远端用户先挂机（来话方向）。 收到RcvCCL事件后，驱动程序会将PendingReason设置为PEND_RemoteHangupOnTalking后，然后将通道状态迁移到**Pending**。如果应用程序在**Connected**状态调用函数SsmSetCalleeHoldFlag开启了“锁定主叫用户”功能，驱动程序会自动向对端交换机发送OPR消息，但状态保持在**Connected**状态。
T1Out	T1 定时器溢出。T1 设置为 60 秒，在通道迁移到 Locked 或 OffHook 状态时自动启动。
T2Out	T2 定时器溢出。T2 设置为 30 秒，在通道迁移到 **Dialing** 状态时自动启动。T2 溢出后驱动程序的操作为： ✧ 设置 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为 ATDL_WaitDialAnsTimeout，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将 PendingReason 设置为 PEND_AutoDialFailed，然后迁移到 Pending 状态。
T4Out	T4 定时器溢出。T4 在配置项MaxWaitAutoDialAnswerTime中设置，在通道迁移到**WaitAnswer**状态时自动启动。T4 溢出后驱动程序的操作为： ✧ 设置 AutoDial 任务的进展值为 DIAL_NOANSWER，失败原因设置为 ATDL_WaitRemotePickupTimeout，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将 PendingReason 设置为 PEND_AutoDialFailed，然后迁移到 Pending 状态。
T5Out	T5 定时器溢出。T5 设置为 20 秒，在通道迁移到 **RcvPhoNum** 状态的 **S_TUP_WaitPrefix** 或 **S_TUP_WaitSAM** 子状态时自动启动。T5 溢出后，驱动程序会自动向对端交换机发送 ADI 消息（表示号码不足），然后迁移到 **Release** 状态。
PhoNumOK	当发生下列事件时，驱动程序会触发本事件： ✧ 驱动程序按照预设的收号规则，完成了全部被叫号码等信息的接收； ✧ 通道处于 **RcvPhoNum** 状态的**_TUP_WaitGSM** 子状态时，设定值为 5 秒的定时器溢出，驱动程序不再等待对端交换机的 GSM 消息。 ✧ 通道处于 **RcvPhoNum** 状态的**_TUP_WaitGSM** 子状态时，收到对端交换机的 GSM 消息。 本事件发生后，如果应用程序通过函数SsmEnableAutoSendKB或配置项AutoSendACM进行开启了“自动应答来话呼叫”功能，驱动程序会自动向对端交换机发送ACM消息，然后迁移到**Ringing**状态。配置项DefaultACM可用来定制驱动程序自动发送的ACM消息中指示被叫用户状态的消息表示语字段，配置项RcvPhoNumCfgLen可以用来设置驱动程序保存被叫号码的方式，函数SsmSetKB可以用来设置ACM消息的后向呼叫表示语中的计费表示语。如果没有开启“自动应答来话呼叫”功能，驱动程序会将PendingReason设置为PEND_WaitBckStpMsg，然后通道迁移到**Suspend**状态，将控制权交给应用程序，由应用程序决定是否接受本次呼叫。
Error	当驱动程序检测到下列情况之一时触发本事件： ➢ 本通道所在数字中继线的 0 时隙上的同步信号丢失（例如线路断开、时钟配置错误等） ➢ TUP 协议栈与 SS7 Server 的之间 TCP/IP 连接断开 ➢ 收到 SS7 Server 的信令不可用消息 当本事件发生时，驱动程序的操作与通道的状态有关，具体为： ➢ **Dialing**、**AppSetCallerID** 或 **WaitAnswer** 状态：驱动程序设置 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos 或 ATDL_Mtp3Unusable，并向应用程序抛出 E_PROC_AutoDial 事件； ➢ **Dialing**、**AppSetCallerID**、**WaitAnswer**、**Connected** 或 Pending 状态：将 PendingReason 设置为 PEND_PcmSyncLos 或 PEND_SsxUnusable，然后迁移到 Pending 状态。
CallFailure	在去话呼叫过程中，如果呼叫失败，驱动程序会触发本事件。当发生下列事件时，去话呼叫失败： ➢ 收到对端交换机的呼叫失败消息。呼叫失败消息包括： ● CFL 消息（呼叫失败）。驱动程序的操作为： ✧ 设置 AutoDial 任务的结果为 DIAL_FAILURE，失败原因设置为 ATDL_RcvCFL，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 驱动程序在设置了 PendingReason 后迁移到 Pending 状态。如果通道处于 WaitAnswer 状态，

	PendingReason 被设置为 PEND_CalleeHangupOnWaitRemotePickUp；如果通道处于 Dialing 状态，PendingReason 被设置为 PEND_AutoDialFailed。 ● SSB、SLB、STB 消息（被叫用户忙）。驱动程序的操作为： ✧ 驱动程序设置 AutoDial 的结果为 DIAL_BUSYTONE，失败原因分别设置为 ATDL_RcvSSB、ATDL_RcvSLB 或 ATDL_RcvSTB，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将 PendingReason 设置为 PEND_AutoDialFailed 后迁移到 Pending 状态。 ● UNN 消息（被叫用户号码为空号）。驱动程序的操作为： ✧ 设置 AutoDial 的结果为 DIAL_INVALID_PHONUM，失败原因设置为 ATDL_RcvUNN，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将 PendingReason 设置为 PEND_AutoDialFailed 后迁移到 Pending 状态。 ● SEC、CGC、NNC、LOS、SST、ACB、DPN、EUM、ADI 消息（其它的呼叫错误）。驱动程序的操作为： ✧ 设置 AutoDial 的结果为 DIAL_FAILURE，失败原因分别设置为 ATDL_RcvSEC、ATDL_RcvCGC、ATDL_RcvNNC 、 ATDL_RcvLOS 、 ATDL_RcvSST 、 ATDL_RcvACB 、 ATDL_RcvDPN 、 ATDL_RcvEUM、ATDL_RcvADI，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将 PendingReason 设置为 PEND_AutoDialFailed 后迁移到 Pending 状态。 a) 发生了 CicReset 事件。 b) 发生了 Blocking 事件。 c) 发生了 Error 事件。

状态转移图中，各状态标识的说明如下表所示：

状态标识	描述
Unusable	通道不可用。
Reseting	电路复位。
BlockedByPBX	对端闭塞状态。对端交换机进行维护时，会向本端发送闭塞消息，以阻止本端对它发起电话呼叫。当通道处于对端闭塞状态时，驱动程序不支持应用程序发出的去话呼叫指令，但仍然可以接受和处理来话呼叫。
BlockedByApp	本端闭塞状态。此时，应用程序不能进行电话呼出操作。本端闭塞通常用于系统维护功能。应用程序可以通过调用函数SsmBlockLocalCh或SsmBlockLocalPCM将通道闭塞，以通知驱动程序阻止电话呼出。**注意：**在本端闭塞状态下，通道仍然可以接受和处理来话呼叫。
Standby	空闲状态。驱动程序初始化成功后，如果通道正常复位，应该进入本状态。应用程序可以在本状态调用SsmAutoDial函数发起去话呼叫业务。 当通道因为呼叫结束而迁移到空闲状态时，如果配置项AutoClearCallerIdBufOnHangup设置为 1，驱动程序会自动清空主叫号码缓冲区。
OffHook	摘机状态。应用程序可以在本状态调用SsmAutoDial函数发起去话呼叫业务。通道进入OffHook状态时，驱动程序启动定时器T1。如果应用程序在T1 溢出之前没有调用SsmAutoDial函数，驱动程序就会将此通道进行回收。
Locked	呼出锁定状态。应用程序可以在本状态调用SsmAutoDial函数发起去话呼叫业务。通道进入本状态时会启动定时器T1。如果应用程序在T1 溢出之前没有调用SsmPickup或SsmAutoDial函数，驱动程序就会将此通道进行回收。
Pending	未决状态。
RcvPhoNum	来话接续过程中的内部状态，由 **S_TUP_WaitSAM**、**S_TUP_WaitPrefix** 和 **S_TUP_WaitGSM** 子状态组成。 驱动程序根据预设的收号规则对收到的被叫号码进行判断，如果符合收号规则，则通道会迁移到后续状态。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。 如果对端交换机发送的 IAM/IAI 消息中包含的被叫号码信息不足，通道会停留在 **S_TUP_WaitSAM** 子状态，每当驱动程序收到对端交换机的 SAO 或 SAM 消息，驱动程序都会将其保存到内部缓冲区中，并向应用程序抛出 E_CHG_RxPhoNumBuf 事件。 当驱动程序按照收号规则收齐了全部被叫号码后，如果需要接收主叫方的信息（如主叫方号码、主叫用户类别等），但对端交换机尚未提供，驱动程序会向对端交换机发送GRQ消息，通道会迁移到**S_TUP_WaitGSM子**状态。配置项ReqTypeIndicators用于定制发送给对端交换机的GRQ消息中的请求类型指示语字段。 当通道收到的被叫号码不符合收号规则时，驱动自动发送 REL（带空号原因值）消息，然后通道迁移到 **Release** 状态。 在 **RcvPhoNum** 状态，应用程序可以：

	✧ 调用SsmHangup/SsmHangupEx函数拒绝本次呼叫。驱动程序会发送REL消息，配置项DefaultHangupRELInd或函数SsmSetIsupFlag（携带ISUP_REL_DENY_SetToOther参数）可用来设REL消息的拒绝原因值。
Ringing	本端振铃状态。应用程序可以用SsmPickup函数接受呼叫，也可以调用SsmHangup函数拒绝呼叫。
Dialing	来话呼叫过程中的状态，由 **S_TUP_WaitDialAnswer** 和 **S_TUP_WaitSetCallerID** 子状态组成。 应用程序调用SsmAutoDial或SsmAutoDialEx函数会引发驱动程序向对端交换机发送IAM消息，通道会迁移到**S_TUP_WaitDialAnswer**子状态。在**S_TUP_WaitDialAnswer**子状态，如果对端交换机希望接收主叫信息，但IAM/IAI消息中没有提供，会向本端发送GRQ消息。驱动程序收到GRQ消息后， ➢ 如果本通道的去话方向的主叫号码缓冲区为空，并且收到的 GRQ 消息表明对端交换机向本端请求主叫号码，则 ✧ 如果配置项AutoSendGSM设置为 1，驱动程序会自动发送GSM消息； ✧ 如果配置项AutoSendGSM设置为 0，通道会迁移到**S_TUP_WaitSetCallerID**子状态，等待应用程序来设置主叫号码等信息，同时启动一个设定值为 2 秒的定时器。如果定时器溢出时应用程序仍然没有调用函数SsmSetTxCallerId，驱动程序会重新迁移到**S_TUP_WaitDialAnswer**子状态。 ➢ 否则，驱动程序主叫号码缓冲区中的字符串作为主叫号码参数，自动发送 GSM 消息
WaitAnswer	等待被叫用户摘机状态，只适用于去话呼叫。此时，被叫方应该听到振铃音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号。
Connected	连接状态，呼叫双方可以进行通话。当通道迁移到 **Connected** 时， ● 如果配置项AlwaysEnableRxDtmf设置为 0，DTMF检测器将被启动； ● 如果配置项AlwaysDetectBargeIn设置为 0， Barge-in 检测器将被启动； ● 信号音检测器将被清零并启动
Release	拆线状态，由 **S_TUP_WaitCLF** 和 **S_TUP_WaitRLG** 子状态组成。

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

状态转移图中，各状态标识在头文件 ShpA3Api.h 中的声明如下表所示：

状态标识	ShpA3Api.h 中的值	ShpA3Api.h 中的宏定义
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
BlockedByPBX	10	S_CALL_RemoteBlock
Unusable	11	S_CALL_UNAVAILABLE
Locked	12	S_CALL_LOCKED
BlockedByApp	20	S_CALL_LocalBlock
Reseting	70	S_TUP_WaitPcmReset
RcvPhoNum	71	S_TUP_WaitSAM
	72	S_TUP_WaitGSM
	74	S_TUP_WaitPrefix
Dialing	75	S_TUP_WaitDialAnswer
	77	S_TUP_WaitSetCallerID
Release	73	S_TUP_WaitCLF
	76	S_TUP_WaitRLG

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

状态转移图中，ChUsable 为驱动程序内部的收号规则判断模块，它的输出描述如下表所示：

模块输出	判决条件

Unusable	如果 Error 事件保持有效，通道不可用
BlockedByApp	Blocking 事件保持有效，并且通道被应用程序闭塞
BlockedByPBX	Blocking 事件保持有效，并且通道被对端交换机闭塞
Standby	Error 事件保持无效

注意：

在此模块，驱动程序还要做下列清理工作：

- ✧ 信号音检测器将被清零并自动关闭；
- ✧ 如果应用程序启动了 WaitDtmf 任务，该任务将被自动终止；
- ✧ 如果应用程序启动了 DTMF 发送任务，该任务将被自动终止；
- ✧ DTMF检测器的缓冲区将被自动清空，如果配置项AlwaysEnableRxDtmf设置为 0， DTMF检测器将被自动关闭；
- ✧ 如果配置项AlwaysDetectBargeIn设置为 0， Barge-in 检测器将被自动关闭；
- ✧ 如果配置项AutoClearCallerIdBufOnHangup设置为 1，存放主叫号码的缓冲区将被自动清空；

1.16.4.5.2通道的闭塞与解除

1.16.4.5.2.1 通道闭塞的概念

（1）通道闭塞

通道被闭塞后，相关通道的呼出（去话呼叫）功能被禁止，但来话接续正常进行。

（2）本地闭塞

相关通道因本方应用程序的指令（函数调用）而阻断本方指定通道的去话业务。

（3）对端闭塞

对端交换机通过向本方发送特定的电路/电路群闭塞消息以阻断本方的去话业务（即阻断对端交换机的来话业务）。对端交换机发送的闭塞消息由驱动程序自动处理。

（4）闭塞对端

应用程序通过向对端交换机发送特定的电路/电路群闭塞消息以阻止对端交换机的去话业务（即阻断本端的来话业务）。

1.16.4.5.2.2 通道闭塞的作用

通道闭塞只适用于数字中继通道。当应用程序因维护等原因需要停机时，需要实施下列 4 个步骤以保证正常的退出顺序：

（1） 闭塞本端所有通道，以阻止本端新的去话业务。

（2） 闭塞对端所有通道，以阻止对端新的去话业务（即本端新的来话业务）。

（3） 等待当前呼叫接续结束。

（4） 程序退出运行。

当通道被闭塞时，驱动程序不再处理新的去话业务（即自动呼出接续相关函数都将返回失败）。

1.16.4.5.2.3 通道闭塞的途径

一个通道可能因下述原因进入闭塞状态：

（1） 收到对端交换机闭塞消息。对于 TUP 协议，通道可能因为收到对端交换机发送的电路闭塞信号 BLO、电路群闭塞信号 SGB、HGB 或 MGB 而闭塞。

（2） 应用程序调用闭塞本地通道的函数 SsmBlockLocalCh() 和 / 或闭塞本地 PCM 的函数 SsmBlockLocalPCM()。

1.16.4.5.2.4 实现通道闭塞的函数列表

实现通道闭塞的函数及其类别如下表：

闭塞类别		函数名称	输出事件
闭塞本端	闭塞通道	SsmBlockLocalCh	
		SsmUnblockLocalCh	
		SsmQueryLocalChBlockState	
	闭塞电路群	SsmBlockLocalPCM	
		SsmUnblockLocalPCM	
		SsmQueryLocalPCMBlockState	
闭塞对端	闭塞通道	SsmQueryOpBlockRemoteCh	
		SsmBlockRemoteCh	收到对端的证实消息（BLA 或 UBA）后抛出 E_CHG_RemoteChBlock 事件
		SsmUnblockRemoteCh	
		SsmGetRemoteChBlockStatus	收到对端的群证实消息（MBA/HBA/SBA 或 MUA/HUA/SUA）后抛出 E_CHG_RemotePCMBlock 事件
	闭塞电路群	SsmBlockRemotePCM	
		SsmUnblockRemotePCM	
		SsmGetRemotePCMBlockStatus	

注：闭塞对端时，本端驱动程序向对端交换机发送的闭塞电路群消息中的范围字段可以通过配置项SendGRMRange进行设置。

1.16.4.5.3 电路的复原

当发生下列事件时，TUP 协议的状态机会自动向对端交换机发送电路群复原消息，将所有的 TUP 通道复原：

- ✧ SS7 信令服务器检测到 MTP3 层提供的服务重新可用时
- ✧ 数字中继线的同步信号恢复时

电路群复原消息中的范围字段可以通过配置项SendGRMRange进行设置。

1.16.4.5.4 “锁定主叫用户”功能

来话呼叫时，当呼叫双方进入通话后，如果主叫方先挂机，对端交换机会向本端发送CCL消息。驱动程序收到CCL消息后，如果应用程序在此之前调用**SsmSetCalleeHoldFlag**函数设置了“锁定主叫用户”功能，驱动将向对端交换机回应OPR消息。

1.16.4.5.5 TUP 协议的配置

使用 TUP 协议时，相关的配置项如下表所示：

配置节	配置项	描述
[BoardId=x]	*PcmNumber*	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作

	PcmSSx	模式、线路的传输类型
	PcmClockMode	
	PcmLinkType	
	UseTS16AsCircuit	设置数字中继线内是否包含 SS7 信令链路，以及具体提供信令链路的时隙编号
	Ss7SignalingTS	
[PcmInfo]	***TotalPcm***	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[SS7]	***AutoHandleTup***	设置是否使用驱动程序提供的呼叫状态机
	Ss7ServerIP	设置 SS7 信令服务器和本机的 IP 地址
	SecondServerIP	
	LocalIP	
[TUP]	MaxWaitAutoDialAnswerTime	去话呼叫：设置工作参数
	CalloutCallerId	
	SetSTSignal	
	AutoSendGSM	
	ConnectReqMsg	
	AddressSignal	
	CalloutIAM_MsgPntr	去话呼叫：定制向对端交换机发送的 IAM 消息
	CalloutIAM_CAT	
	CallingIndicatorBit	
	ReqTypeIndicators	
	OriginalCalleeAddrInd	
	DefaultACM	来话呼叫：设置工作参数
	AutoSendACM	
	ReqTypeIndicators	
	RingToPending	
	RcvPhoNumCfgLen	来话呼叫：设置收号规则
	DefaultRcvPhoNumLen	
	DefaultRcvCallerID	
	MaxPhoNumRule	
	Rule	
	HangupRingSendCBK	来话呼叫：设置拒绝来话呼叫时使用的消息类型

注：***XXXX*** 字样为必须配置的项目，XXXX 为可选的配置项目

1.16.4.6 ISUP 协议的编程与应用

1.16.4.6.1ISUP 协议的配置

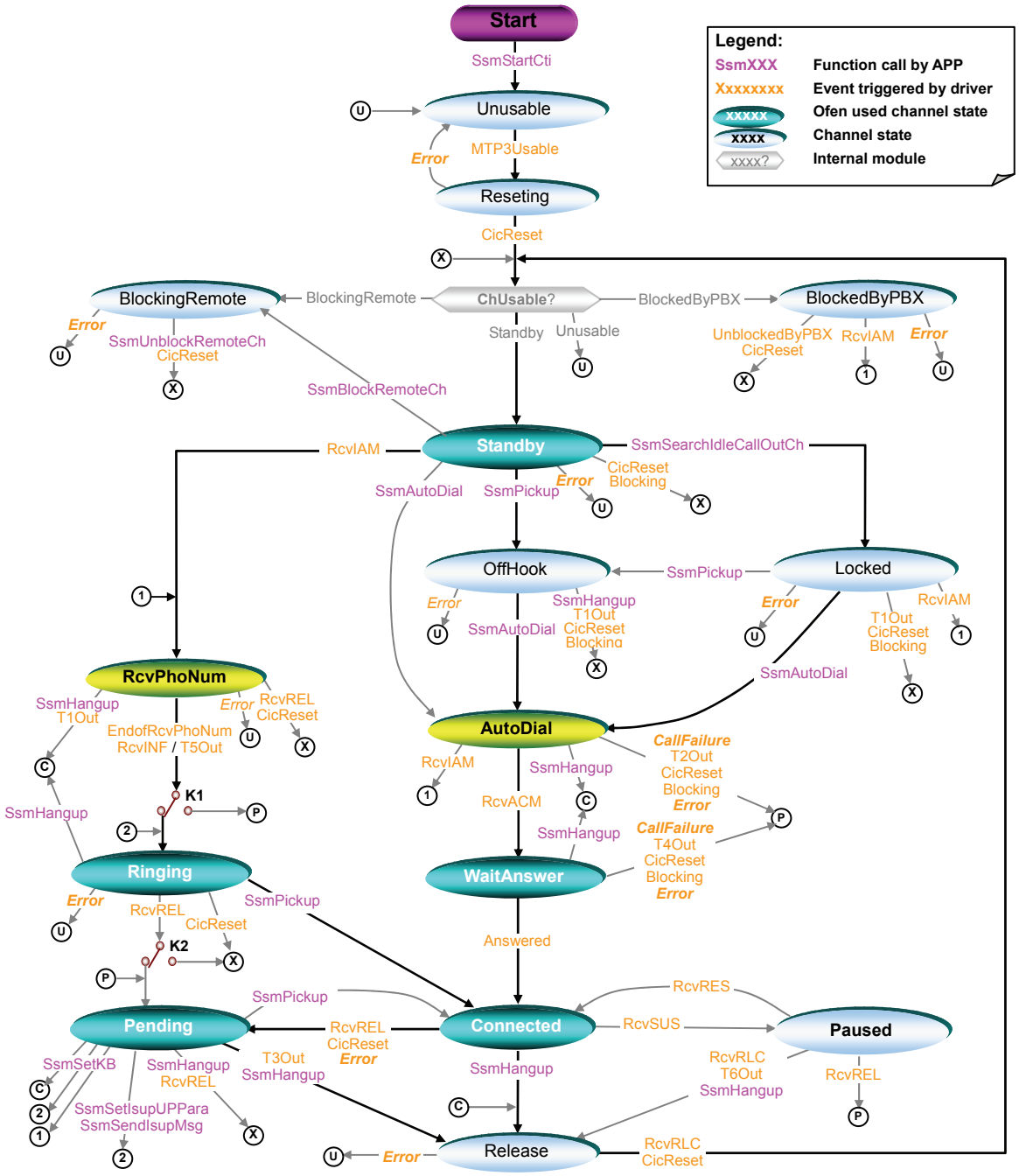
使用 ISUP 协议时，相关的配置项如下表所示：

配置节	配置项	描述
[BoardId=x]	***PcmNumber***	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	PcmSSx	
	PcmClockMode	
	PcmLinkType	
	UseTS16AsCircuit	设置数字中继线内是否包含 SS7 信令链路，以及具体提供信令链路的时隙编号
	Ss7SignalingTS	
[PcmInfo]	***TotalPcm***	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[SS7]	***TotalIsupPcm***	设置本机中使用 ISUP 协议的数字中继线
	IsupPcm	
	AutoHandleIsup	设置是否使用驱动程序提供的呼叫状态机
	Ss7ServerIP	设置 SS7 信令服务器和本机的 IP 地址

	SecondServerIP	
	LocalIP	
[ISUP]	MaxWaitAutoDialAnswerTime	去话呼叫：设置工作参数
	CalloutCallerId	
	SetSTSignal	
	DefaultNatureOfConnectionInd	去话呼叫：定制向对端交换机发送的 IAM 消息
	DefaultIAM_ForwardCallInd	
	DefaultIAM_CAT	
	DefaultIAM_TransmissionMediumRequirment	
	DefaultIAM_CalleeParam	
	DefaultIAM_CallerParam	
	DefaultIAM_OriginalCalleeParam	
	bSubscriberSI，SubscriberSI	
	bOptionalFCI，OptionalFCI	
	Usr2UsrInfo	
	AutoSendINF	去话呼叫：设置应答 INF 消息的模式
	RingToPending	来话呼叫：设置工作参数
	AutoSendACM	
	DefaultCalledPickupMsg	
	DefaultBackwardCallInd	
	DefaultHangupRELInd	
	DefaultCauseInd	
	SaveRGNTo1stPhoNumStr	
	DefaultRcvPhoNumLen	来话呼叫：设置收号规则
	DefaultRcvCallerID	
	MaxPhoNumRule	
	Rule	
	AddressSignal	设置 ISUP 协议状态机的工作参数

注：***XXXX*** 字样为必须配置的项目，XXXX 为可选的配置项目

1.16.4.6.2ISUP 通道的状态转移



上图中，驱动程序产生的内部事件如下表所述：

事件名称	描述
Mtp3Usable	驱动程序收到 SS7 Server 的信令可用消息后，会向所有通道的状态机发送本事件，使通道从 **Unusable 状态**迁移到 **Reseting** 状态，开始电路复原。
CicReset	当驱动程序收到电路复原消息（GRA、GRS 或 RSC）时触发本事件。 本事件发生后，如果通道处于**AutoDial**或**WaitAnswer**状态，去话呼叫失败，驱动程序会向应用程序抛出E_PROC_AutoDial事件，函数SsmGetAutoDialFailureReason可以用来获取AutoDial失败的具体原因。
Blocking	本通道被本端闭塞或被对端闭塞： ● 对端闭塞：驱动程序收到对端交换机的群闭塞消息（CGB）或电路闭塞消息（BLO）

<table>
<tr><td></td><td>● 本端闭塞：应用程序调用SsmBlockLocalCh或SsmBlockLocalPCM函数。</td></tr>
<tr><td>UnblockedByPBX</td><td>解除对本端电路的闭塞：驱动程序收到对端交换机的 CGU、UBL 消息时触发本事件。</td></tr>
<tr><td>RcvIAM</td><td>驱动程序收到对端交换机的来话呼叫建立请求消息（IAM）。
驱动程序会自动保存最新收到的IAM原始消息，函数SsmGetIsupUPPara（携带C_ISUP_IAM参数）可以用来获取此原始消息，SsmGetRedirectionInfReason、SsmGetKA可以分别获取IAM消息里的转呼原因值字段和“主叫方类别”参数。
驱动程序是否接受此来话呼叫请求，取决于通道的状态：
<table>
<tr><th>通道状态</th><th>描述</th></tr>
<tr><td>BlockedByPBX</td><td>在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫，因此，驱动程序会对 IAM 消息予以响应</td></tr>
<tr><td>AutoDial</td><td>如果在去话呼叫接续过程中收到对端交换机的呼叫建立请求消息IAM，表明发生了“同抢”。根据ISUP协议规定，如果通道的CIC编号为本端主控的电路，则IAM消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫，本次去话呼叫失败，驱动程序会向应用程序抛出E_PROC_AutoDial事件，函数SsmGetAutoDialFailureReason可以用来获取AutoDial失败的具体原因。</td></tr>
<tr><td>OffHook Locked</td><td>通道正在准备进行去话呼叫，但尚未送出 IAM 消息。此时收到对端的来话呼叫请求，驱动程序会对 IAM 消息予以响应</td></tr>
<tr><td>Standby</td><td>接受来话呼叫</td></tr>
<tr><td>其它</td><td>对此 IAM 消息予以拒绝</td></tr>
</table>
如果驱动程序接受来话呼叫，在将 IAM 消息中的被叫号码信息保存到缓冲区后，向应用程序抛出 E_CHG_RxPhoNumBuf 事件，然后将通道迁移到 RcvPhoNum 状态。</td></tr>
<tr><td>EnfofRcvPhoNum</td><td>驱动程序根据预设的收号规则完成全部被叫号码的接收。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。</td></tr>
<tr><td>RcvINF</td><td>驱动程序收到对端交换机的 INF 消息。INF 消息包含了主叫方的相关信息。</td></tr>
<tr><td>RcvREL</td><td>驱动程序收到对端交换机的拆线消息（REL）。函数SsmGetReleaseReason可以获取具体的释放原因。
当本事件发生时，如果通道处于
● RcvPhoNum 状态：
表明在来话呼叫的建立过程中，对端交换机（主叫方）取消了本次呼叫。
● Ringing 状态：
表明在来话呼叫的建立过程中，当本端处于振铃状态时，对端交换机取消了本次呼叫。应用程序可以通过对K2开关的控制来决定是否干预通道的状态迁移。K2 由配置项RingToPending决定。如果配置项RingToPending设置为 1，驱动程序会设置PendingReason为PEND_RemoteHangupOnRinging，然后将通道迁移到Pending状态，以等待应用程序的进一步处理；否则，驱动程序开始拆除本次来话呼叫。
● Connected 状态：
表明主叫用户已经挂机。驱动程序会将 PendingReason 设置为 PEND_RemoteHangupOnConnected，然后将通道状态会迁移到 Pending。</td></tr>
<tr><td>RcvRLC</td><td>驱动程序收到对端交换机的 RLC 消息，拆线过程结束</td></tr>
<tr><td>RcvACM</td><td>驱动程序收到对端交换机的 ACM 消息（被叫地址全）。
在去话呼叫接续过程中，当应用程序调用SsmAutoDial函数启动一次去话呼叫任务后，驱动程序会向对端交换机发送IAM消息。当对端交换机完成被叫号码等必要信息的接收后，会向本端发送ACM消息，指示被叫用户的空闲状态。驱动程序收到ACM消息后，会将原始的ACM消息保存到内部缓冲区中，函数SsmGetIsupUPPara（携带C_ISUP_ACM参数）可以用来获取此消息，然后将AutoDial任务的进展设置为DIAL_ECHOTONE，并将通道迁移到WaitAnswer状态。函数SsmGetKB可以用来获取此ACM消息中后向呼叫表示语字段(共 2 个字节)中的低 8 位比特。
对端交换机在发送了ACM消息后，可能会继续发送CPG消息（呼叫进展）。函数SsmGetCpg可以取出原始的CPG消息。</td></tr>
<tr><td>Answered</td><td>在去话呼叫过程中，对端交换机完成被叫号码的接收后，如果被叫用户处于空闲状态，会向被叫用户发出振铃信号。如果被叫用户摘机，对端交换机会向本端发送应答消息（ANM）。
驱动程序收到 ANM 消息后，将 AutoDial 任务的进展设置为 DIAL_VOICE，AutoDial 任务终止，并向应用程序抛出 E_PROC_AutoDial 事件。</td></tr>
<tr><td>RcvSUS</td><td>收到对端交换机的 SUS（暂停）消息。</td></tr>
<tr><td>RcvRES</td><td>收到对端交换机的 REC（恢复）消息。</td></tr>
<tr><td>CallFailure</td><td>在去话呼叫过程中，当收到对端交换机的REL拆线等消息时，去话呼叫失败，驱动程序会向应用程序抛出E_PROC_AutoDial事件，函数SsmGetAutoDialFailureReason可以用来获取AutoDial失败的具体原因。</td></tr>
<tr><td>T1Out</td><td>T1 定时器溢出。T1 设置为 60 秒，在通道迁移到 Locked 或 OffHook 状态时自动启动。</td></tr>
</table>

T2Out	T2 定时器溢出。T2 设置为 25 秒，在通道迁移到**AutoDial**状态中的**S_ISUP_WaitDialAnswer**子状态时自动启动。如果 T2 溢出，本次去话呼叫失败，驱动程序会向应用程序抛出 E_PROC_AutoDial 事件，函数 SsmGetAutoDialFailureReason可以用来获取AutoDial失败的具体原因。
T3Out	T3 定时器溢出。通道进入 Pending 状态时启动，设定值为 60 秒。
T4Out	T4 定时器溢出。T4 在配置项MaxWaitAutoDialAnswerTime中设置，在通道迁移到**WaitAnswer**状态时自动启动。如果T4 溢出，驱动程序会向应用程序抛出E_PROC_AutoDial事件，函数SsmGetAutoDialFailureReason可以用来获取AutoDial失败的具体原因。
T5Out	T5 定时器溢出。T5 设置为 10 秒，在通道迁移到 **RcvPhoNum** 状态的 **S_ISUP_WaitINF** 子状态时自动启动。T5 溢出后，驱动程序不再等待对端交换机的 INF 消息。
T6Out	T6 定时器溢出。T6 设置为 180 秒，在通道迁移到 Suspend 状态时自动启动
Error	当驱动程序检测到下列情况之一时触发本事件： ✧ 本通道所在数字中继线的 0 时隙上的同步信号丢失（例如线路断开、时钟配置错误等） ✧ ISUP 协议栈与 SS7 Server 之间的 TCP/IP 连接断开 ✧ 收到 SS7 Server 的信令不可用消息 当本事件发生时，如果通道处于**AutoDial**或**WaitAnswer**状态，去话呼叫失败，驱动程序会向应用程序抛出E_PROC_AutoDial事件，函数SsmGetAutoDialFailureReason可以用来获取AutoDial失败的具体原因。

各状态标识的说明如下表所示：

状态标识	描述
Unusable	“不可用”状态。
Reseting	“电路复位”状态。
BlockingRemote	“闭塞对端交换机”状态。 在 **BlockingRemote** 状态，对端交换机不能发起去话呼叫，但本端可以发起去话呼叫。
BlockedByPBX	“被对端交换机闭塞”状态。 对端交换机进行维护时，会向本端发送闭塞消息，以阻止本端对它发起电话呼叫。当通道处于对端闭塞状态时，驱动程序不支持应用程序发出的去话呼叫指令，但仍然可以接受和处理来话呼叫。
Standby	“空闲”状态。 驱动程序初始化成功后，如果通道正常复位，应该进入本状态。 应用程序可以在本状态调用SsmAutoDial函数发起去话呼叫业务。 当通道因为呼叫结束而迁移到空闲状态时，如果配置项AutoClearCallerIdBufOnHangup设置为 1，驱动程序会自动清空主叫号码缓冲区。
Locked	“呼出锁定”状态。 应用程序可以在本状态直接调用SsmAutoDial函数发起去话呼叫业务。 通道迁移到本状态时，驱动程序会启动定时器T1。如果应用程序在T1 溢出之前没有调用SsmPickup或SsmAutoDial函数，驱动程序就会将此通道进行回收。T1 定时器的设定值为 60 秒。
OffHook	“摘机”状态。 应用程序可以在本状态调用SsmAutoDial函数发起去话呼叫业务。 通道进入OffHook状态时，驱动程序会启动定时器T1。如果应用程序在T1 溢出之前没有调用SsmAutoDial函数，驱动程序就会将此通道进行回收。
AutoDial	去话接续过程中的内部状态，由 **S_ISUP_WaitDialAnswer** 状态和 **S_ISUP_WaitSetCallerID** 状态组成。 应用程序调用 SsmAutoDial 或 SsmAutoDialEx 函数后会引发驱动程序向对端交换机发送 IAM 消息，并会迁移到 **S_ISUP_WaitDialAnswer** 子状态。在 **S_ISUP_WaitDialAnswer** 子状态，如果对端交换机希望接收主叫信息，但 IAM 消息中没有提供，会向本端发送 INR 消息。驱动程序收到 INR 消息后， ✧ 如果驱动程序内部的去话方向的主叫号码缓冲区不为空，驱动程序会将该主叫号码信息装配到 INF 消息中，并自动将 INF 消息回送给对端交换机，同时保持状态不变； ✧ 如果驱动程序内部的去话方向的主叫号码缓冲区为空，并且配置项AutoSendINF设置为 1，驱动程序以配置项CalloutCallerId或函数SsmSetTxCallerId设置的主叫信息作为参数，自动应答INF消息，并保持状态不变； ✧ 如果驱动程序内部的去话方向的主叫号码缓冲区为空，并且配置项AutoSendINF设置为 0，通道会迁移到**S_ISUP_WaitSetCallerID**子状态，同时启动T3 定时器。在**S_ISUP_WaitSetCallerID**子状态，如果应用程序调用了SsmSetTxCallerId函数，驱动程序会将设置的主叫号码信息装配到INF消息中，并将INF消息回送

	给对端交换机。如果直到T3 定时器溢出时，应用程序仍然没有调用SsmSetTxCallerId函数，驱动程序会以配置项CalloutCallerId 的设定值作为参数，自动向对端发送INF消息，然后将通道迁移到**S_ISUP_WaitDialAnswer**子状态。
WaitAnswer	“等待被叫用户摘机”状态，只适用于去话呼叫。 此时，被叫方应该听到振铃音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号。
Pending	“挂起”状态。函数SsmGetPendingReason可用于获取具体的挂起原因。 ➢ 如果 PendingReason 为 PEND_WaitBckStpMsg，表明来话呼叫的接续过程已经完成，正等待应用程序接受或拒绝来话呼叫。应用程序可以： ✧ 调用SsmSetKB函数接受或拒绝来话呼叫。 ◆ 如果KB=1/6/7：驱动程序会向对端交换机发送ACM消息，接受来话呼叫。如果应用程序在调用SsmSetKB函数之前已经调用SsmSetIsupUPPara函数(携带C_ISUP_ACM参数)向驱动程序提交了一个自行构建的ACM消息，该ACM消息将被发送给对端交换机；否则，驱动程序会使用自动构建一个ACM消息。发送完ACM消息后，通道会迁移到**S_CALL_RINGING**状态； ◆ 如果 KB=2/3/4/5/9，驱动程序会向对端交换机发送 REL 消息，拒绝来话呼叫，REL 消息中携带的拒绝原因值分别为被叫忙/地址不全/呼叫拒绝/无应答/用户缺席，通道开始拆除连接，迁移到 **Release** 状态； ◆ 如果 KB=8，驱动程序会向对端交换机发送 INR 消息，通道的状态会重新回到 **RcvPhoNum** 状态。 ✧ 顺序调用SsmSetIsupUPPara函数(携带C_ISUP_ACM参数)和SsmSendIsupMsg函数接受来话呼叫，然后通道会迁移到**S_CALL_RINGING**状态。 ✧ 调用SsmHangup或SsmHangupEx函数，拒绝本次来话呼叫，驱动程序开始拆除连接，通道会迁移到**Release**状态。更多信息请参见相关函数的说明。 ✧ 调用SsmPickup函数接受来话呼叫。驱动程序会向对端交换机发送CON消息，然后通道会迁移到**Connected**状态。 **注意：如果驱动程序在本状态收到对端交换机的拆线消息（REL），驱动程序会自动发送 RLC 消息，然后将通道迁移到 Standby 状态** ➢ 无论是因为什么原因导致通道进入**Pending**状态，应用程序都可以调用SsmSetIsupFlag（携带ISUP_PhoNumREL参数），命令驱动程序向对端交换机发送包含号码改发信息的拆线消息（REL），并使通道迁移到**Release**状态。 ➢ 如果 PendingReason 为： ✧ PEND_RemoteHangupOnTalking（来话方向，进入通话后检测到主叫用户先挂机）； ✧ PEND_CalleeHangupOnTalking（去话方向，进入通话后检测到被叫用户先挂机）； ✧ PEND_RemoteHangupOnRinging（来话方向，本端振铃时，主叫方取消呼叫） 应用程序可以调用SsmHangup或SsmHangupEx函数挂机。
RcvPhoNum	来话接续过程中的内部状态，由 **S_ISUP_Suspend** 和 **S_ISUP_WaitINF** 子状态组成。 驱动程序根据预设的收号规则对收到的被叫号码进行判断，如果符合收号规则，则通道会迁移到后续状态。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。 ➢ 在来话呼叫过程中，K1 开关控制“自动应答来话呼叫”功能。当通道完成了全部被叫号码以及其它相关信息的接收后， ✧ 如果开启了“自动应答来话呼叫”功能，驱动程序会自动向对端交换机发送ACM消息，然后迁移到**Ringing**状态。配置项DefaultBackwardCallInd可用来设置驱动程序自动发送的ACM消息中的后向呼叫表示语，函数SsmSetKB可以用来设置ACM消息的后向呼叫表示语中的计费表示语。 ✧ 如果没有开启“自动应答来话呼叫”功能，驱动程序会将 PendingReason 设置为 PEND_WaitBckStpMsg，然后通道迁移到 **Suspend** 状态，将控制权交给应用程序，由应用程序决定是否接受本次呼叫。 K1 开关可通过函数SsmEnableAutoSendKB或配置项AutoSendACM进行设置，缺省值为开启。 ➢ 每当驱动程序收到对端交换机的 SAM 消息，驱动程序都会将其保存到内部缓冲区中，并向应用程序抛出 E_CHG_RxPhoNumBuf 事件。 ➢ 当通道收到被叫号码不符合收号规则时，驱动自动发送 REL（带空号原因值）消息，然后通道迁移到 **Release** 状态。 ➢ 应用程序也可调用SsmHangup/SsmHangupEx函数拒绝本次呼叫，默认发送带拒绝原因值的REL消息，配置项DefaultHangupRELInd 设置挂机发送 REL 消息的原因值，或调用函数 SsmSetIsupFlag 参数 nType=ISUP_REL_DENY_SetToOther(=100)改变默认原因值。
Ringing	“振铃”状态。 应用程序可以： ✧ 调用SsmPickup或SsmPickupANX函数接受呼叫。配置项DefaultCalledPickupMsg用于选择驱动程序向对端交换机发送的消息类型（ANM或CON），配置项DefaultBackwardCallInd可以设置ACM和CON消息中的后向

	呼叫指示码； ✧ 调用SsmHangup或SsmHangupEx函数拒绝本次呼叫。 如果通道处于**Ringing**状态时，驱动程序收到对端交换机的REL消息（即状态图中的RcvRLC事件），表明对端交换机取消了本次呼叫。此时，K2 开关可以用来控制通道的状态迁移。K2 由配置项RingToPending决定。如果配置项RingToPending设置为 1，驱动程序会设置PendingReason为PEND_RemoteHangupOnRinging，然后将通道迁移到**Pending**状态，以等待应用程序的进一步处理；否则，驱动程序立即开始自动拆除本次来话呼叫。
Connected	“连接”状态，呼叫双方可以进行通话。 当通道迁移到 **Connected** 状态时，驱动程序会 ✧ 如果配置项AlwaysEnableRxDtmf设置为 0，自动启动DTMF检测器； ✧ 如果配置项AlwaysDetectBargeIn设置为 0，自动启动Barge-in 检测器； ✧ 清零并启动信号音检测器 应用程序可以： ✧ 调用任何语音处理的相关函数； ✧ 随时调用调用SsmHangup或SsmHangupEx函数终止本次呼叫。 在 **Connected** 状态，如果驱动程序收到对端交换机的拆线消息（状态图中的 RcvREL 事件），表明通话时对端用户已经挂机。
Paused	“暂停”状态。
Release	“拆线”状态。 在 **Release** 状态，驱动程序会等待对方发送 RLC 消息（释放完成消息）

上表中，状态的值与宏定义均在文件 **ShpA3Api.h** 中声明，具体如下标所示：

状态标识	ShpA3Api.h 中声明	
	值	宏
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_CONNECTED
Pending	7	S_CALL_PENDING
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Unusable	11	S_CALL_UNAVAILABLE
Locked	12	S_CALL_LOCKED
Release	121	S_ISUP_WaitRLC
Reseting	122	S_ISUP_WaitReset
BlockingRemote	123	S_ISUP_LocallyBlocked
BlockedByPBX	124	S_ISUP_RemotelyBlocked
Paused	129	S_ISUP_Suspend
RcvPhoNum	120	S_ISUP_WaitSAM
	126	S_ISUP_WaitINF
AutoDial	125	S_ISUP_WaitDialAnswer
	127	S_ISUP_WaitSetCallerID

驱动程序内部的逻辑判断模块 **ChUsable** 用于判断通道是否可用。它的判断逻辑与输出的关系为：

模块输出	判决条件
Unusable	如果 Error 事件保持有效，通道不可用
BlockedByPBX	通道被对端交换机闭塞
BlockingRemote	通道上保持对对端交换机的闭塞
Standby	Error 事件保持无效

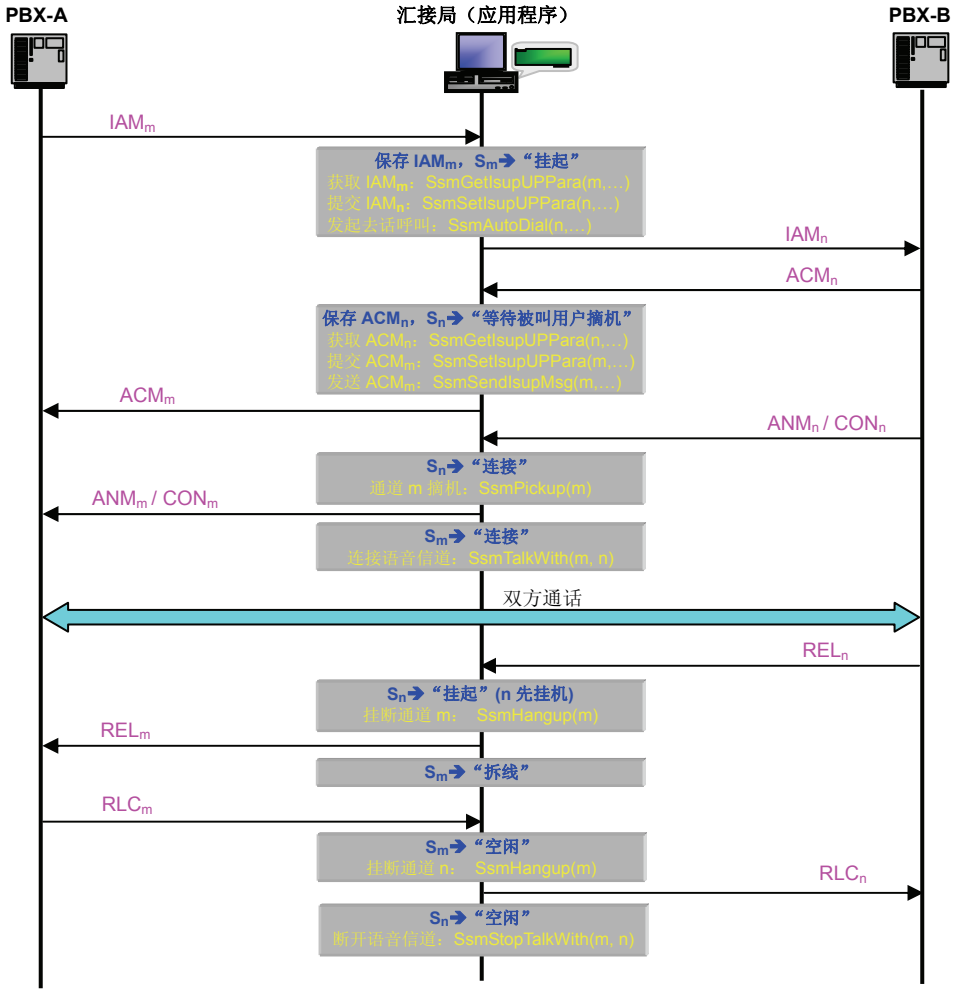
注意：在 ChUsable 模块中，驱动程序还要自动执行下列操作：

- **信号音检测器将被清零并自动关闭；**
- **如果应用程序启动了 WaitDtmf 任务，该任务将被自动终止；**
- **如果应用程序启动了 DTMF 发送任务，该任务将被自动终止；**
- **DTMF检测器的缓冲区将被自动清空，如果配置项AlwaysEnableRxDtmf设置为 0， DTMF检测器将被自动关闭；**
- **如果配置项AlwaysDetectBargeIn设置为 0， Barge-in 检测器将被自动关闭；**
- **如果配置项AutoClearCallerIdBufOnHangup设置为 1，存放主叫号码的缓冲区将被自动清空；**

1.16.4.6.3ISUP 通道工作于汇接局方式

通常情况下，使用 ISUP 协议的 SHD 系列板卡都工作在端局模式下。如果希望 ISUP 通道工作在汇接局模式下，可以按照下列方式实现：

假设应用程序中，通道 m 在 PBX-A 上的电路编号也为 m，并且通道 m 的“自动应答来话呼叫”功能为关闭状态；通道 n 在 PBX-B 上的电路编号为 n。ISUP 通道工作于汇接局方式的转接过程如下图所示：



上图中，

蓝色字体部分表示驱动程序的内部操作；

S_m、S_n 分别表示通道 m 和通道 n 的状态；

XXX_m、XXX_n 分别表示应用程序与 PBX-A 和 PBX-B 之间的 ISUP 消息；

黄色字体部分为应用程序的动作。

1.16.4.6.4不使用 SynCTI 提供的 ISUP 协议状态机

如果应用程序希望自行实现ISUP协议的状态机，可以通过修改配置项AutoHandleIsup实现，更多信息请参见“SS7 的高级编程”部分内容。

1.16.4.7 SS7 的高级编程

1.16.4.7.1使用 MTP3 层提供的服务

应用程序也可以直接使用 SynCTI 驱动程序提供的 MTP3 接口功能，来自行实现 SS7 第 4 层的任何协议。如果下列配置项设置为 0：

- ✧ AutoHandleTup （对于TUP协议）
- ✧ AutoHandleIsup （对于ISUP协议）
- ✧ AutoHandleSccp （对于SCCP协议）

每当驱动程序从SS7 服务器收到一条消息，都不会被送到驱动程序内部的状态机，而是存入一个缓冲区进行缓存，同时抛出E_RCV_Ss7Msu事件。应用程序收到E_RCV_Ss7Msu事件后，可以调用函数SsmGetSs7Msu取出消息，然后进行后续处理。函数SsmSendSs7Msu可以用来发送一条消息。

每当SS7 服务器到某个DPC的路由状态发生变化时（可用，或不可用），会向SynCTI驱动程序发送相关消息，从而引发驱动程序向应用程序抛出E_CHG_Mtp3State事件。对于单个DPC的应用场合，函数SsmGetMtp3State可以获取到达该DPC的信令路由是否可用；如果应用系统连接到了多个DPC，函数SsmGetMtp3StateEx可以获取到达指定DPC的信令路由是否可用。函数SsmGetMtp2Status可以用来获取指定的 64Kbps的信令链路的工作状态。

1.16.4.7.2使用 SS7 服务器的客户端软件编程接口

对于不具有七号信令功能的设备，可以使用基于 SHD 系列板卡的 SS7 信令服务器，通过 TCP/IP 与该设备连接，使其具有 SS7 信令处理能力。此类应用系统需要使用 SS7 信令服务器提供的客户端软件编程接口。SS7 信令服务器提供的客户端软件编程接口以动态连接库的形式提供，其内部封装了 TCP/IP 通讯模块和 SS7 信令处理功能，向应用程序提供透明的 SS7 信令功能，更多信息请参见《SS7 信令服务器的客户端软件编程接口》。

1.16.4.7.3使用 SS7 服务器的虚电路编程接口

虚电路是指 SS7 电路交换中不占用语音资源的电路，一般应用于预付费系统。

SynCTI 驱动程序通过将配置项 Pcm[k]=i,j 中的参数 i 指定为-1，来设定没有与某个物理数字中继线绑定的虚拟数字中继线，通过 SS7 信令服务器中的 ss7server.ini 配置文件中[ISUPRouter]节来绑定虚拟数字中继线及目的信令点码。有关虚电路的 API 函数的详细信息请参见文档《SynCTI 程序员手册-SS7 信令虚电路的编程.doc》。

1.16.4.7.4使用 SCCP 协议的编程接口

SynCTI驱动程序提供了SCCP协议的高级编程接口，更多信息请参见文档中的说明。如果应用程序需要自行实现SCCP，需要将配置项AutoHandleIsup设置为 0，并使用SynCTI驱动程序提供的MTP3 层服务即可。更多信息请参见本章中“使用MTP3 层提供的服务”部分内容。

1.16.5 ISDN 通道的编程

1.16.5.1 基本概念

1.16.5.1.1网络侧模式与用户侧模式

在 ISDN 协议中，由于通过数字中继线连接两端的协议并不完全对等，因此，在 ISDN 协议中有“用户侧”和“网络侧”的区别。用户侧指用户终端，即指与 ISDN 交换机相连的一端；网络侧是指与 ISDN 用户终端相连的另外一端。在本文档中，将 SynCTI 驱动程序中工作于用户侧模式的数字中继线称为用户侧数字中继线，将工作于网络侧模式的数字中继线称为网络侧数字中继线。

1.16.5.1.2TEI 值

TEI：终端设备标识符，用于标识连接在服务接入点内点到点数据链路连接中的终端。TEI 子字段允许规定 128 个 TEI 值，因此，允许的取值范围为 0~127，其中，0~63 表示固定的 TEI 值（由设备选择），表明 L2 接收/发送 I 帧或 UI 帧所使用的 TEI 不变；64~126 表示动态的 TEI 值，表明 L2 接收/发送 I 帧或 UI 帧所使用的 TEI 可以变化（由网络选择）；127 只适用于广播链路。

配置项UserTEIValue、NetTEIValue分别用于配置用户侧、网络侧的TEI值。

1.16.5.1.3通路识别表示方法

在 ISDN 信令的呼叫过程中，用来指示每次呼叫所使用的通道信息的内容，称为通路识别。通路识别主要用于识别信令程序所控制的接口范围内的通道。通路识别有下列 2 种表示方法：

- ✧ 时隙图方式：通路识别信息用 32 Bits（4 个字节）来表达，分别对应 32 个时隙，bit=1 表示该时隙被使用，bit=0 表示未被使用。实际应用中，通常只有 1 个比特被置 1，其余比特置 0。
- ✧ 号码方式：通路识别信息用 8 Bits（1 个字节）来表达，最高位 Bit7 固定为 1，其余 7 个比特（即 Bit6~Bit0）的值即表示相应的时隙编号。

配置项NetChIdentify用于设置网络侧的某个链路呼出时通路识别表示方式；配置项UserChIdentify用于设置用户侧的某个链路呼出时通路识别表示方式。

1.16.5.1.4端局方式与汇接局方式

电话网分为本地电话网和长途电话网。本地电话网是在同一编号区内的网络，由端局、汇接局和传输链路组成；

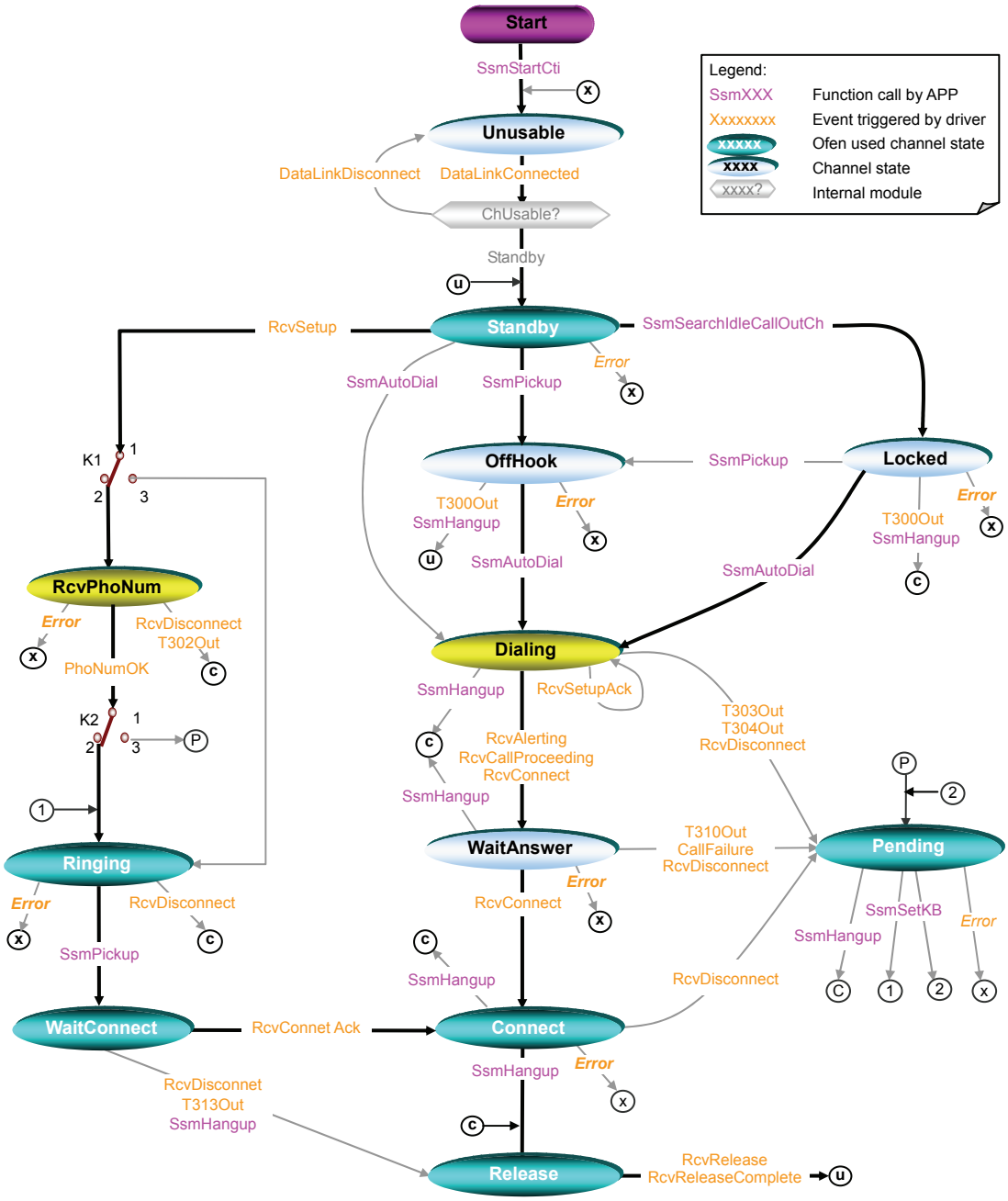
长途电话网是在不同的编号区之间通话的网络，由长途交换局和传输链路组成。目前，电话交换局是电话网中的核心，采用数字程控交换设备，每一路电话编码为 64Kbit/s 的数字信号，占据一次群中的某一时隙，在信令的控制下进行时隙交换，从而和各个不同的用户相连。根据服务区域的大小，电话交换局可以分为一级中心、二级中心、三级中心、四级中心和五级中心，即 C1、C2、C3、C4 和 C5。其中 C1、C2、C3、C4 为长途汇接局，C5 为端局。随着电话网的数字化进程的实现，C1、C2 合并为一级，即 DC1，C3、C4 合并为一级，即 DC2，中国的电话网从五级网演变为三级网，一级交换中心之间形成网状连接。

SynCTI驱动程序提供的ISDN协议状态机可以既可以工作在端局方式，也可以工作在汇接局方式，可通过配置项UserIsReceivePhoNum（用户侧）或NetIsReceivePhoNum（网络侧）进行设置。在处理来话呼叫时，如果驱动程序工作于端局方式，会根据预设的收号规则接收被叫号码等信息；如果工作于汇接局模式，则预设收号规则将不起作用，任何号码都可以呼入到本端。有关收号规则的更多信息请参见本章中“设置来话呼叫的收号规则”部分内容。

1.16.5.1.5ISDN 号码与 ISDN 地址

ITU-T 把 ISDN 号码和 ISDN 地址分开。ISDN 号码是与 ISDN 网络及 ISDN 编号方案有关的号码，它包含了足够的信息，使网络能够为呼叫选择正确的路由。一个 ISDN 地址包括 ISDN 号码及必备的和/或可选的附加寻址信息，这个附加信息并不是用来供 ISDN 选择呼叫路由，而是供用户将呼叫分配到合适的终端。

1.16.5.2 ISDN 通道的状态转移图



状态转移图中，驱动程序自动触发的内部事件如下表示：

事件	事件描述
DataLinkConnected	当数据链路层与对端交换机的连接建立后，会触发本事件。驱动程序会向应用程序抛出 E_CHG_ISDNStatus 事件。
DataLinkDisconnected	当数据链路层与对端交换机的连接被断开。驱动程序会向应用程序抛出 E_CHG_ISDNStatus 事件。
RcvSetup	收到对端交换机的 SETUP（呼叫建立请求）消息。 驱动程序会根据通道的当前状态，判断是否接受本次来话呼叫： ✧ **Standby**状态：接受此来话呼叫。驱动程序会将SETUP消息中的被叫号码等信息保存到内部缓冲区，并向应用程序抛出E_CHG_RxPhoNumBuf事件。如果驱动程序工作在汇接局方式，则来话呼叫的收号规则将被忽略，任何号码都可以被驱动程序接受，驱动程序会将通道直接迁移到**Ringing**状态。K1 开关是用来是否以汇接局的方式进行接续。K1 的状态可以通过配置项UserIsReceivePhoNum（用户侧）或NetIsReceivePhoNum（网络侧）来设置，更多信息请参见本章中“端局方式与汇接局方式”部分内容。

	✧ 其它状态：丢弃此 SETUP 消息。
PhoNumOK	驱动程序根据预设的收号规则，完成全部号码（包括主叫号码和被叫号码）的接收。
RcvDisconnect	收到对端交换机的 DISCONNECT 消息。 当本事件发生时，如果通道处于 ✧ **RcvPhoNum**、**WaitAnswer**、**Ringing** 状态：表明在来话呼叫的建立过程中，对端交换机（主叫方）取消了本次呼叫； ✧ **Connect** 状态：对于来话呼叫，在 **Connect** 状态收到 DISCONNECT 消息，表明对端交换机先挂机。驱动程序会向对端交换机发送 RELEASE 消息，在将 PendingReason 设置为 PEND_ ISDN_CALLOVER 后，将通道迁移到 **Pending** 状态； ✧ **Dialing**状态：表明去话呼叫失败，具体的失败原因可以通过函数过SsmGetAutoDialFailureReason获得。
RcvConnectAck	收到对端交换机的 CONNECT ACK 消息。本端向对端交换机发送 CONNECT 消息后，如果收到 CONNECT ACK 消息，表示连接建立。
RcvConnect	收到对端交换机的 CONNECT 消息。如果通道处于 **WaitAnswer** 状态，驱动程序会自动向对端交换机应答 CONNECT ACK 消息。
RcvAlerting	收到对端交换机的 ALERTING 消息。此时，被叫方已经进入振铃状态。
RcvCallProceeding	收到对端交换机的 CALL PROCEEDING 消息。此时，被叫方已经进入振铃状态。
RcvSetupAck	收到对端交换机的 SETUP ACK 消息。本事件不会影响通道的状态迁移。
RcvRelease	收到对端交换机的 RELEASE 消息。驱动程序会自动向对端交换机应答 RELEASE COMPLETE 消息，并开始拆除连接。
RcvReleaseComplete	收到对端交换机的 RELEASE COMPLETE 消息。本端向对端交换机发送 RELEASE 消息后，如果收到 RELEASE COMPLETE 消息，表示连接拆除完毕。
T300Out	T300 定时器溢出，T300 设置为 60s，搜索到空闲通道和摘机后操作的时间限制，以保证通道的最大利用率。用户侧通过配置项UserDialTime、网络侧通过NetDialTime来设置溢出时间。
T302Out	来话呼叫时，当驱动程序收到对端交换机的 SETUP 消息后，会根据预设的收号规则对 SETUP 消息中的被叫方号码进行检查。如果号码没有收齐，驱动程序会将通道的状态迁移到 **RcvPhoNum** 状态的 S_ISDN_IN_RCVING_NO 子状态，并启动定时器 T302。如果 T302 溢出，本次来话呼叫失败，驱动程序会自动向对端交换机发送 RELEASE 消息，并将通道状态迁移到 **Standby**。 T302 可以通过配置项UserT302（用户侧）或NetT302（网络侧）进行设置。
T313Out	来话呼叫时，当通道进入**Ringing**状态后，如果应用程序调用SsmPickup函数摘机，驱动程序会向对端交换机自动发送CONNECT消息，并迁移到**WaitConnect**状态，同时启动定时器T313。如果T313 溢出，本次来话呼叫失败，驱动程序会向对端交换机发送DISCONNECT消息。 T313 可以通过配置项UserT313进行设置，但仅适用于用户侧。
T303Out	去话呼叫时，当本端送出 SETUP 消息后，驱动程序会启动定时器 T303。如果 T303 溢出时，驱动程序还没有从对端交换机收到 ALERTING、CALL PROCEEDING、SETUP ACK 或 CONNECT 消息，驱动程序会重新发送一次 SETUP 消息，并重新启动 T303。如果 T303 再次溢出，本次去话呼叫失败，驱动程序会向对端交换机发送 RELE COMP 消息，并将通道迁移到 **Pending** 状态。 T303 可以通过配置项UserT303（用户侧）或NetT303（网络侧）进行设置。
T304Out	去话呼叫时，当本端送出SETUP消息后，如果对端交换机不能从SETUP消息中获得全部需要的被叫号码等信息，会向本端发送SETUP ACK消息。本端收到此SETUP ACK消息后，会等待应用程序追加被叫号码，同时会启动定时器T304。此时，应用程序可以调用函数SsmAppendPhoNum追加被叫号码。驱动程序收到新的号码后，会向对端交换机发送INFO消息，并重新启动T304。如果T304 溢出时，驱动程序还没有从对端交换机收到CALL PROCEEDING、ALERTING、CONNECT或DISCONNECT消息，本次去话呼叫失败，驱动程序会向对端交换机发送DISCONNECT消息，并将通道迁移到**Pending**状态。 T304 可以通过配置项UserT304（用户侧）或NetT304（网络侧）进行设置。
T310Out	去话呼叫时，当本端送出 SETUP 消息、并且从对端交换机收到 CALL PROCEEDING 消息后，驱动程序会启动定时器 T310。如果 T310 溢出时，驱动程序还没有从对端交换机收到 ALERTING 或 CONNECT 消息，本次去话呼叫失败，驱动程序会向对端交换机发送 DISCONNECT 消息，并将通道迁移到 **Pending** 状态。 T310 可以通过配置项UserT310（用户侧）或NetT310（网络侧）进行设置。
Error	当驱动程序检测到下列情况之一时触发本事件： ✧ 本通道所在数字中继线的 0 时隙上的同步信号丢失（例如线路断开、时钟配置错误等）； ✧ 从对端交换机收到错误消息； ✧ 数据链路层无法建立对端交换机的连接； 驱动程序的动作： ✧ 在 **Dialing**、**WaitAnswer** 状态：去话呼叫失败。驱动程序会设置 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 其它状态： 通道将迁移到不可用状态

各状态标识的说明如下表所示：

状态标识	描述
Unusable	通道“不可用”状态。 ✧ 如果 DatalinkDisconnect、Error 事件之一保持有效，通道不可用，输出 Unusable； ✧ 其它情况，通道空闲，输出 Standby。
Standby	“空闲”状态。 驱动程序初始化成功后，如果ISDN的数据链路层顺利建立到对端交换机的连接，应该进入本状态。应用程序可以在本状态调用SsmAutoDial函数发起去话呼叫业务。当通道因为呼叫结束而迁移到本状态时，如果配置项AutoClearCallerIdBufOnHangup设置为 1，驱动程序会自动清空主叫号码缓冲区。
OffHook	“摘机”状态。 通道进入**OffHook**状态时，驱动程序会启动定时器T300。如果应用程序在T300 溢出之前没有调用SsmAutoDial函数，驱动程序会将此通道进行回收，将通道重新迁移到空闲状态。
Locked	“呼出锁定”状态。 应用程序调用SsmSearchIdleCallOutCh函数时，驱动程序会将函数返回的通道临时锁定，同时启动定时器T300。如果应用程序在T300 溢出之前没有调用SsmPickup或SsmAutoDial函数，驱动程序就会将此通道进行回收。
Dialing	去话呼叫过程中的中间状态，由 **S_ISDN_OUT_WATI_NET_RESPONSE** 和 **S_ISDN_OUT_PLS_APPEND_NO**两个子状态组成。应用程序调用SsmAutoDial 函数后，驱动程序会： ✧ 向对端交换机发送 SETUP 消息； ✧ 同时启动 T303 定时器； ✧ 将通道迁移到 **S_ISDN_OUT_WATI_NET_RESPONSE** 子状态，以等待对端交换机的响应信号； ✧ 向应用程序抛出 E_PROC_AutoDial 事件。 在**S_ISDN_OUT_WATI_NET_RESPONSE**子状态，如果收到对端交换机的SETUP ACK消息，表明对方需要本端提供进一步的号码信息，驱动程序会将通道迁移到**S_ISDN_OUT_PLS_APPEND_NO**子状态，同时启动定时器T304。此时，应用程序可以调用SsmAppendPhoNum追加被叫号码。
WaitAnswer	“等待被叫用户摘机”状态。通道迁移到本状态时，驱动程序会启动定时器 T310。 函数SsmGetFlag（携带F_ISDNNet_WaitRemotePickup参数）可以用来获取通道迁移到本状态的具体原因。
WaitConnect	等待对端交换机应答 CONNECT ACK 消息。通道迁移到本状态时，驱动程序会启动定时器 T313。
Connect	“连接”状态，呼叫双方可以进行通话。 当通道迁移到 **Connected** 时： ✧ 如果配置项AlwaysEnableRxDtmf 设置为 0，DTMF检测器将被启动； ✧ 如果配置项AlwaysDetectBargeIn设置为 0， Barge-in 检测器将被启动； ✧ 信号音检测器将被清零并启动； 驱动内部响应事件： ✧ RcvDisconnect 事件：表明通话时对端先挂机，驱动程序在将挂起原因设置为 ISDN_CALLOVER 后迁移到“挂起”状态； ✧ Error 事件：表明通话过程中遇到 Error 事件之一，通道将迁移到“不可用”状态。
Ringing	“振铃”状态。 通道完成了全部被叫号码以及其它相关信息的接收。
Pending	“挂起”状态，等待应用程序的进一步指令。
RcvPhoNum	来话呼叫时的内部状态，由 **S_ISDN_IN_CHK_CALL_IN** 和 **S_ISDN_IN_RCVING_NO** 子状态组成。 收到对端交换机的 SETUP 消息后，如果 SETUP 消息中包含了本端收号所需要的全部信息，驱动程序会触发内部事件 PhoNumOK；否则，驱动程序会向对端交换机发送 SETUP ACK 消息，并将通道迁移到 **S_ISDN_IN_CHK_CALL_IN** 子状态，等待对端交换机发送进一步的号码信息。在 **S_ISDN_IN_CHK_CALL_IN** 子状态，如果收到对端交换机的 INFO 消息，驱动程序会将通道迁移到 **S_ISDN_IN_RCVING_NO** 子状态。 如果收到的被叫号码不符合收号规则，驱动程序会自动向对端交换机发送 RELEASE 消息，然后将通道迁移到 **S_CALL_STANDBY** 状态。 配置项组合DefaultRcvPhoNumLen、DefaultRcvCallerID、RcvPhoNumCfgLen或DefaultRcvPhoNumLen、MaxPhoNumRule、Rule可以用来设置收号规则。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。 通道完成收号过程后，迁移的下一个状态取决于开关K2。K2 开关用来决定是由驱动程序自动应答ALERTING 消息，还是由应用程序接管。K2 的状态可以通过配置项 UserSideAutoSendAck（用户侧）或 NetSideAutoSendAck（网络侧）或函数SsmEnableAutoSendKB进行设置。如果设置为由驱动程序自动应答，驱动程序在向对端交换机发送ALERTING后，将通道迁移到**Ringing**状态；否则，驱动程序会启动一个定时器，

	定时器的时长由配置项UserSideDefaultAckTimer（用户侧）或NetSideDefaultAckTimer（网络侧）进行设置，在将PendingReason设置为PEND_WaitBckStpMsg后，驱动程序会将通道迁移到**Pending**状态，将控制权交给应用程序，由应用程序决定是否接受本次呼叫。如果定时器溢出，驱动程序会根据配置项UserSideDefaultAck（用户侧）或NetSideDefaultAck（网络侧）的设置，自动向对端交换机发送应答消息。
Release	正在拆除连接。

各状态标识在头文件 ShpA3Api.h 中的声明如下表所示：

状态标识	ShpA3Api.h 中的值	ShpA3Api.h 中的宏定义
Unusable	11	S_CALL_UNAVAILABLE
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Dialing	81	S_ISDN_OUT_WAIT_NET_RESPONSE
	82	S_ISDN_OUT_PLS_APPEND_NO
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Connect	3	S_CALL_TALKING
WaitConnect	85	S_ISDN_IN_WAIT_TALK
Ringing	2	S_CALL_RINGING
Pending	7	S_CALL_PENDING
Locked	12	S_CALL_LOCKED
RcvPhoNum	83	S_ISDN_IN_CHK_CALL_IN
	84	S_ISDN_IN_RCVING_NO
Release	88	S_ISDN_WAIT_HUANGUP

驱动程序内部逻辑判断模块的描述如下表所示：

模块输出	判决条件
Unusable	如果 Error、DataLinkDisconnect、Circuit Reset 之一事件保持有效，通道不可用。
Standby	Error 事件保持无效、DataLinkConnect 事件保持有效、电路重起成功，通道状态空闲。

注意：在此模块中，驱动程序还要做下列清理工作：

- 清零并关闭信号音检测器；
- 如果应用程序启动了 WaitDtmf 任务，该任务将被自动终止；
- 如果应用程序启动了 DTMF 发送任务，该任务将被自动终止；
- DTMF检测器的缓冲区将被自动清空，如果配置项AlwaysEnableRxDtmf设置为 0， DTMF检测器将被自动关闭；
- 如果配置项AlwaysDetectBargeIn设置为 0， Barge-in 检测器将被自动关闭；
- 如果配置项AutoClearCallerIdBufOnHangup设置为 1，存放主叫号码的缓冲区将被自动清空；

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_PROC_AutoDial	应用程序调用SsmAutoDial函数即启动了AutoDial任务。当AutoDial任务有进展时，驱动程序向应用程序抛出本事件。
E_CHG_RxPhoNumBuf	在来话呼叫过程中，每当驱动程序收到对端交换机的被叫号码信息驱动程序向应用程序抛出本事件
E_CHG_ISDNStatus	驱动程序收到 ISDN 协议数据链路层可用消息触发本事件，并向应用程序抛出本事件

注：事件的声明在文件 ShpA3Api.h 中。

1.16.5.3 ISDN 协议的配置

使用 ISDN 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	***PcmNumber***	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	PcmSSx	

	PcmClockMode	
	PcmLinkType	
[PcmInfo]	***TotalPcm***	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[ISDN]	***AutoHandleIsdn***	设置是否使用驱动程序提供的呼叫状态机
	UseISDNMode	ISDN 用户侧、网络侧协议设置
	TotalUserLinker	设置本机中 ISDN 用户侧协议的数字中继线的总数，以及中继线的逻辑编号与物理编号之间的映射关系
	UserPcmLink	
	UserTEIValue	ISDN 用户侧协议基本参数设置
	UserRestarTime	
	UserEstablishTime	
	UserT302	
	UserCrcMode	
	UserStatusReason	
	UserDialTime	ISDN 用户侧去话呼叫：设置工作参数
	UserT310	
	UserChIdentify	ISDN 用户侧去话呼叫：定制向对端交换机发送的 SETUP 消息
	UserCalledNoSet	
	UserCallingNoSet	
	UserNumIsFull	
	UserChPreference	
	UserHighLayerCompatible	
	UserLowLayerCompatible	
	CalloutCallerId	
	UserSideAutoSendAck	ISDN 用户侧来话呼叫：设置工作参数
	UserSideDefaultAckTimer	
	UserIsReceivePhoNum	
	UserVoiceFormat	ISDN 用户侧硬件编码设置
	TotalNetLinker	设置本机中 ISDN 网络侧协议的数字中继线的总数，以及中继线的逻辑编号与物理编号之间的映射关系
	NetPcmLink	
	NetTEIValue	ISDN 网络侧协议基本参数设置
	NetRestarTime	
	NetEstablishTime	
	NetT302	
	NetCrcMode	
	NetStatusReason	
	NetDialTime	ISDN 网络侧去话呼叫：设置工作参数
	NetT310	
	NetChIdentify	ISDN 网络侧去话呼叫：定制向对端交换机发送的 SETUP 消息
	NetCalledNoSet	
	NetCallingNoSet	
	NetNumIsFull	
	NetChPreference	
	NetHighLayerCompatible	
	NetLowLayerCompatible	
	CalloutCallerId	
	NetSideAutoSendAck	ISDN 网络侧来话呼叫：设置工作参数
	NetSideDefaultAckTimer	
	NetIsReceivePhoNum	
	DefaultRcvCallerID	
	NetVoiceFormat	ISDN 网络侧硬件编码设置
	DefaultRcvPhoNumLen	ISDN 用户侧、网络侧来话呼叫：收号规则设制
	MaxPhoNumRule	
	Rule	

	UserStatusReason NetStatusReason	ISDN 用户侧、网络侧的高级选项
	DebugSwitch	ISDN 用户侧、网络侧日志输出设置
	DecodeUserSideIsdnMsg	
	DecodeNetSideIsdnMsg	
	RecordIFrame	
	LogMode	

注：***XXXX*** 字样为必须配置的项目，XXXX 为可选的配置项目

1.16.5.4 ISDN 高级编程接口

SynCTI 驱动程序还提供了高级编程接口，输出 HDLC 层的原始消息，由应用程序自己处理 ISDN 消息。有关 ISDN 高级编程接口的配置项和函数如下表所示：

类型	名称	描述
配置项	AutoHandleIsdn	设置是否使用 ISDN 的高级编程接口
函数	SsmSendIsdnMSU	发送一条消息
	SsmCheckIsdnMSU	查询是否有消息到达
	SsmGetIsdnMSU	取出一条 ISDN 消息

1.16.6 SS1 通道编程指南

1.16.6.1 选择信令协议

SynCTI支持No.1 信令和Line side协议，可通过配置项ProtocolType进行设置。如果使用No.1 信令，配置项mfcr2_Protocol用于选择国别。

1.16.6.2 设置接续过程的基本工作参数

配置项TxCas_CD用于设置本端向对端发送ABCD信令码时C/D码元的值。配置项RxCASFilterTime用于设置对端交换机送出的ABCD信令的最小保持时间，以消除线路上的信号抖动，适用于信号质量比较差的E1/T1 线路。

R2 信号的最小保持时间可以通过配置项RxR2FilterTime或函数SsmSetFlag（携带F_RXR2FILTERTIME参数）来进行设置。

当应用程序退出时，配置项IsBlockSS1In用于设置是否向对端交换机发送闭塞命令，以通知对端交换机不要再向本端发起呼叫。

1.16.6.3 设置状态机中通道的工作方式

配置项EnableAutoCall 用于设置是否使用SynCTI驱动程序提供的呼叫状态机，函数SsmSetAutoCallDirection或配置项AutoCallInTimeSlot用于设置通道的呼叫方向，如可以将通道设置成只处理来话呼叫或只处理去话呼叫。

1.16.6.4 中国 1 号信令的状态机描述

1.16.6.4.1来话呼叫时启用号码拦截功能

“被叫号码拦截”功能，常用于一些关键的部门，是为了减少主叫方的误拨号对应用系统造成的影响。例如，对于采用 Synway 板卡构建的 110 报警系统，如果主叫方拨打的是“110”，这是一个合法的被叫号码，应用程序应该处理；如果主叫方拨打的是“1101”，报警系统就不应该处理本次来话呼叫。

“被叫号码拦截”功能及工作模式由配置项PhoNumHoldup或函数SsmSetFlag（携带F_RCVPHONUMHOLDUP参数）控制，其工作参数可以通过配置项A1ToA3pWaitTime和A3pTime进行设置。例如，一个 110 接警系统可以如

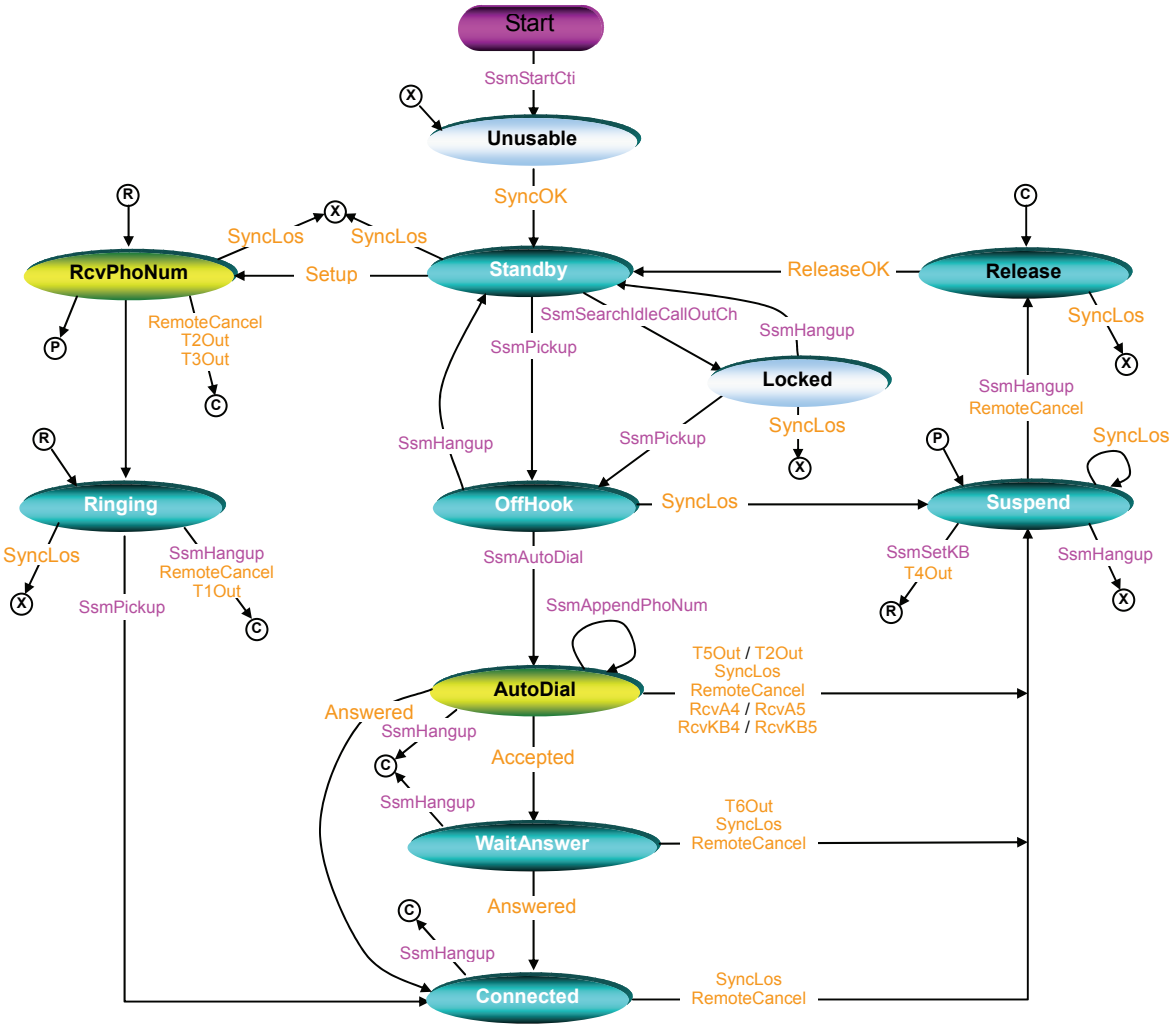
下设置拦截 110x形式的被叫号码：

PhoNumHoldup=1
A1ToA3pWaitTime=1000
A3pTime=150

1.16.6.4.2与 Dialogic 公司 SS1 通道对接

使用Synway板卡的SS1 通道Dialogic公司的SS1 通道进行对接呼叫时，如果Synway板卡作为呼入端，需要对配置项ToRingingDelayTime进行设置。

1.16.6.4.3中国 1 号信令的状态转移



上图中，内部事件的描述如下表所示：

事件名称	描述
RemoteCancel	对端交换机取消了呼叫
SyncLos	数字中继线 0 时隙上的同步信号丢失。

SyncOK	检测到数字中继线 0 时隙上的同步信号。
Setup	收到对端交换机的来话呼叫请求
T1Out	T1 定时器溢出。T1 在通道迁移到“振铃”状态时启动。T1 的设定值为 75 秒。
T2Out	T2 定时器溢出。SS1 通道在来话呼叫过程中，需要等待对端交换机的应答信号。T2 设置最大等待时间，如果 T2 溢出，表明本次互控过程失败。 T2 可以通过配置项MaxWaitMfcTime进行设置。
T3Out	T3 定时器溢出。SS1 通道在来话呼叫过程中，当本端完成被叫号码的接收后，会向对端交换机发送 A3 信号，同时启动 T3，等待 KD 信号。如果 T3 溢出时对端交换机还没有送出 KD 信号，来话呼叫接续失败。 T3 可以通过配置项MaxWaitKDTime进行设置。
T4Out	T4 定时器溢出。SS1 通道在来话呼叫过程中，当本端完成被叫号码的接收后，如果配置项AutoSendKB设置为 0，表明应用程序希望执行决定是否应答本次呼叫，通道会进入“挂起”状态，并启动定时器T4。如果T4 溢出时，应用程序还没有调用SsmSetKB函数对呼叫进行应答，驱动程序会拒绝本次来话呼叫。 T4 可以通过配置项MaxWaitSetKBTime进行设置。
T5Out	T5 定时器溢出。 应用程序调用 SsmAutoDial 函数发起去话呼叫后，驱动程序会向对端交换机发送“占用”请求信号，并等待对端交换机的证实信号。如果 T5 溢出时，仍然没有收到对端交换机的证实信号，本次去话呼叫失败。 T5 可以通过配置项MaxWaitOccupyAckTime进行设置。
T6Out	T6 定时器溢出。 应用程序调用 SsmAutoDial 函数发起去话呼叫后，如果被叫用户空闲，通道会启动 T6。如果 T6 溢出时，被叫用户仍然没有摘机，本次去话呼叫失败。 T6 可以通过配置项MaxWaitAutoDialAnswerTime进行设置。
RcvA4	去话接续过程中收到对端交换机的 A4 信号（机键拥塞，接续尚未到达被叫之前遇到空号），去话呼叫失败
RcvA5	去话接续过程中收到对端交换机的 A5 信号（空号），去话呼叫失败
RcvKB4	去话接续过程中，收到对端交换机的 KB 信号为 4（用户忙或机键拥塞），去话呼叫失败
RcvKB5	去话接续过程中，收到对端交换机的 KB 信号为 5（被叫用户为空号），去话呼叫失败
Accepted	去话接续过程中，对端交换机完成被叫号码的接收后，向本端发送了 KB 信号值为 1 或 6，表明被叫用户空闲
Answered	去话接续过程中，被叫用户接受了呼叫
ReleaseOK	完成拆除与对端交换机的连接

各状态的说明如下表所示：

状态标识	描述
Standby	空闲状态。驱动程序初始化成功后，如果通道正常复位，应该进入本状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。 当通道因为呼叫结束而迁移到空闲状态时，如果配置项AutoClearCallerIdBufOnHangup设置为 1，驱动程序会自动清空主叫号码缓冲区。
OffHook	“摘机”状态。 如果在“摘机”状态发生 SyncLos 事件，驱动程序会： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 PEND_PcmSyncLos；
Ringing	本端振铃状态。应用程序可以： ✧ 调用函数SsmGetCallerId获取主叫方的信息 ✧ 调用SsmPickup函数接受呼叫，也可以调用SsmHangup函数拒绝呼叫。
Connected	连接状态，呼叫双方可以进行通话。当通道迁移到 Talking 时，驱动程序会： ✧ 启动 DTMF 检测器； ✧ 启动 Barge-in 检测器； ✧ 清零并启动信号音检测器 驱动程序对事件的响应为： ➢ RemoteCancel 事件： 表明通话时对端先挂机，驱动程序在将挂起原因设置为 PEND_RemoteHangupOnTalking 后迁移到“挂起”状态； ➢ SyncLos 事件：

	✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 PEND_PcmSyncLos；
Pending	等待应用程序进一步的指示。 驱动程序对事件的响应为： ➢ T4Out 事件： 驱动程序自动将 KB 信号设置为 4（机键拥塞），然后恢复来话接续过程。 ➢ 应用程序调用SsmSetKB函数： 如果挂起原因为 PEND_WaitBckStpMsg，通道将恢复来话接续过程。 ➢ SyncLos 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 PEND_PcmSyncLos； ➢ 应用程序调用SsmHangup函数： 如果挂起原因为 PEND_PcmSyncLos，通道状态将迁移到“不可用”；其它的挂起原因则进入“拆线”状态。
WaitAnswer	等待被叫用户摘机。此时，被叫方应该听到振铃音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号。 驱动程序对事件的响应为： ➢ T6Out 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_NOANSWER，失败原因设置为 ATDL_WaitRemotePickupTimeout，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_NOANSWER； ➢ SyncLos 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 PEND_PcmSyncLos； ➢ RemoteCancel 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_SS1RcvCAS_HANGUP，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_DIALING_BWD_HANGUP；
Unusable	“通道不可用”。
Locked	“呼出锁定”状态。
Release	“拆除连接”状态。
RcvPhoNum	来话接续过程中的内部状态，由多个状态组成。驱动程序根据预设的收号规则对收到的被叫号码进行判断，如果符合收号规则，则通道会迁移到后续状态。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。 ➢ 在来话呼叫过程中，当通道完成了全部被叫号码以及其它相关信息的接收后，需要向对端交换机发送 KB 信号， ✧ 如果开启了“自动应答来话呼叫”功能，驱动程序会将配置项MfcKB的设定值作为KB信号发送给对端交换机，然后迁移到Ringing状态； ✧ 如果没有开启“自动应答来话呼叫”功能，驱动程序会将 PendingReason 设置为 PEND_WaitBckStpMsg，然后通道迁移到 Pending 状态，将控制权交给应用程序，由应用程序决定是否接受本次呼叫。 “自动接受来话”开关可通过函数SsmEnableAutoSendKB或配置项AutoSendKB进行设置，缺省值为开启。 ➢ 每当驱动程序收到主叫方的一个号码，驱动程序都会抛出 E_CHG_RxPhoNumBuf 事件。
AutoDial	去话接续过程中的内部状态，由多个状态组成。 **去话呼叫过程中，应用程序可以设置的参数包括：** ➢ 本端需要向对端交换机提供的KA信号（主叫类别），可以通过函数SsmSetKA或配置项MfcKA进行设置； ➢ 本端需要向对端交换机提供的KD信号（发端业务类别），可以通过函数SsmSetKD或配置项MfcKD进行设置； ➢ 如果对端交换机需要本端提供主叫方电话号码，驱动程序会在收到A6 信号、发送完KA信号后会自动将保存在缓冲区中的本端电话号码发送给对端交换机。本端的电话号码可通过函数SsmSetTxCallerId或配置项TxCallerId进行设置； **如果去话呼叫失败，驱动程序会依据内部事件的类型做下列处理：** ➢ SyncLos 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件；

	✧ 将挂起原因设置为 PEND_PcmSyncLos； ➢ T5Out 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_NO_DIALTONE，失败原因设置为 ATDL_SS1WaitOccupyAckTimeout，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_NOBWDACK； ➢ RemoteCancel 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_SS1RcvCAS_HANGUP，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_DIALING_BWD_HANGUP； ➢ RcvA4 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_BUSYTONE，失败原因设置为 ATDL_SS1RcvA4，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_BWD_A4； ➢ RcvKB4 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_BUSYTONE，失败原因设置为 ATDL_SS1RcvKB4，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_BWD_KB4； ➢ RcvA5 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_INVALID_PHONUM，失败原因设置为 ATDL_SS1RcvA5，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_BWD_A5； ➢ RcvKB5 事件： ✧ 将 AutoDial 任务的进展值设置为 DIAL_INVALID_PHONUM，失败原因设置为 ATDL_SS1RcvKB5，并向应用程序抛出 E_PROC_AutoDial 事件； ✧ 将挂起原因设置为 SS1OUT_BWD_KB5； ➢ T2Out 事件： 将 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为下列值之一： ◆ ATDL_SS1WaitAxTimeout ◆ ATDL_SS1WaitAxStopTimeout ◆ ATDL_SS1WaitAxTimeoutOnTxCallerId ◆ ATDL_SS1WaitAxStopTimeoutOnTxCallerId ◆ ATDL_SS1WaitKBTimeout ◆ ATDL_SS1WaitKBStopTimeout 对应的挂起原因分别设置为 ◆ SS1OUT_TIMEOUT_BWD_A ◆ SS1OUT_TIMEOUT_BWD_A_STO ◆ SS1OUT_TIMEOUT_CALLERID_BWD_A1 ◆ SS1OUT_TIMEOUT_CALLERID_BWD_A1_STOP ◆ SS1OUT_TIMEOUT_BWD_KB ◆ SS1OUT_TIMEOUT_BWD_KB_STOP； 并向应用程序抛出 E_PROC_AutoDial 事件。

状态标识、状态值之间的对应关系如下表所示：

状态标识	值	ShpA3Api.h 对应的宏定义
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Unusable	11	S_CALL_UNAVAILABLE
Locked	12	S_CALL_LOCKED
Release	36	S_CALL_SS1WaitIdleCAS

RcvPhoNum	—	S_CALL_Ss1InWaitPhoNum S_CALL_Ss1InWaitFwdStop S_CALL_Ss1InWaitCallerID S_CALL_Ss1InWaitKD S_CALL_Ss1InWaitKDStop S_CALL_SS1WaitIdleCAS S_CALL_SS1PhoNumHoldup S_CALL_Ss1InWaitStopSendA3p
AutoDial	—	S_CALL_Ss1OutWaitBwdAck S_CALL_Ss1OutTxPhoNum S_CALL_Ss1OutWaitAppendPhoNum S_CALL_Ss1OutTxCallerID S_CALL_Ss1OutWaitKB S_CALL_Ss1OutDetectA3p

状态机的输出事件类型如下表所示：

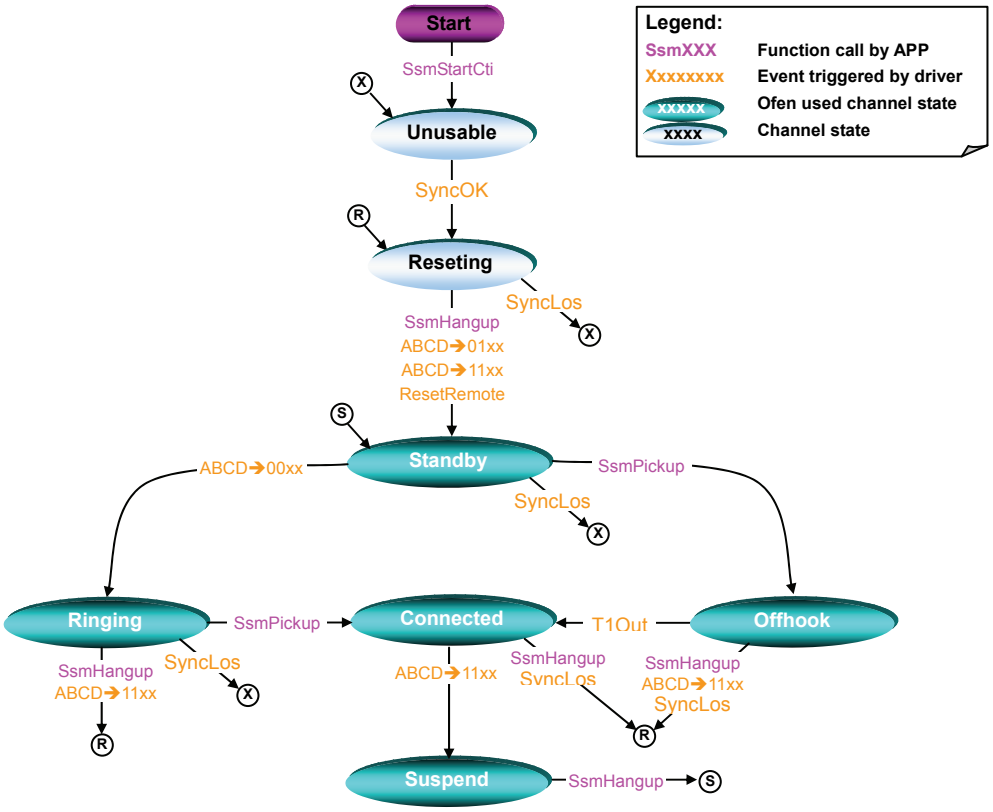
事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件。
E_CHG_RxPhoNumBuf	在来话呼叫过程中，每当驱动程序收到对端交换机的被叫号码信息，驱动程序向应用程序抛出本事件。
E_RCV_R2	当应用程序调用SsmGetR2函数获得R2 消息时，如果驱动程序收到对端的新的R2 信号值，驱动程序向应用程序抛出本事件。
E_RCV_CAS	当应用程序调用 SsmGetCAS 函数取得指定通道上发送 ABCD 信令的当前值时，如果驱动程序收到了当前的 CAS 值，驱动程序向应用程序抛出本事件。
E_CHG_CIDExBuf	驱动程序收到 CID 扩展接收缓冲区有变化后，驱动程序向应用程序抛出本事件。应用程序得到该事件后，可调用 SsmGetCallerIdEx 函数取得 CID 扩展接收缓冲区内容。

1.16.6.4.4输出呼叫过程中的调试信息

配置项MfcR2ToRxCallerIdBuf用于控制是否将呼叫过程中对端交换机发送的R2 信号保存到主叫号码扩展缓冲区中，以便于对呼叫过程进行观察。函数SsmGetCallerIdEx用于取出该缓冲区中的字符串。

配置项Ss1OutputLog用于控制是否将接收和发送的ABCD位码输出到日志文件Ss1Output.log中。

1.16.6.5 Line side 协议的状态机描述



上图中，内部事件的描述如下表所示：

事件名称	描述
SyncLos	数字中继线 0 时隙上的同步信号丢失。
SyncOK	检测到数字中继线 0 时隙上的同步信号。
ABCD→00xx	对端交换机的 ABCD 位码变化为 00xx
ABCD→01xx	对端交换机的 ABCD 位码变化为 01xx
ABCD→10xx	对端交换机的 ABCD 位码变化为 10xx
ResetRemote	驱动程序完成对对端交换机的电路复位操作，即向对端交换机发送空闲指示的ABCD位码。具体发送的空闲指示的ABCD位码的值由配置项Ss1SendIdleState决定。
T1Out	T1 定时器溢出。T1 定时器在通道迁移到“摘机”状态时启动。T1 可以通过配置项LSWaitPickup进行设置。

各状态的说明如下表所示：

状态标识	描述
Standby	空闲状态。驱动程序初始化成功后，如果通道正常复位，应该进入本状态。如果应用程序在本状态调用SsmPickup，驱动程序会向对端交换机发送ABCD=11xx。
OffHook	“摘机”状态。 在“摘机”状态，应用程序可以向对端交换机发送被叫号码等信息，如何向对端交换机发送被叫号码等信息，取决于交换机的类型，详细信息请向交换机厂商查询。通道进入“摘机”状态时，会启动定时器 T1。
Ringing	本端“振铃”状态。 在“振铃”状态，应用程序可以调用SsmPickup函数接受呼叫，也可以调用SsmHangup函数拒绝呼叫。
Connected	“连接”状态，呼叫双方可以进行通话。 通道迁移到“连接”状态时，驱动程序会自动开启 DTMF 检测器。
Suspend	“挂起”状态。 需要注意的是，驱动程序不会设置挂起原因，因此，应用程序调用SsmGetPendingReason函数没有任何意义。

Unusable	“通道不可用”。
Reseting	“电路复位”状态。

状态标识、状态值之间的对应关系如下表所示：

状态标识	值	ShpA3Api.h 对应的宏定义
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_TALKING
Suspend	7	S_CALL_PENDING
Unusable	11	S_CALL_UNAVAILABLE
Reseting	36	S_CALL_SS1WaitIdleCAS

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件。

1.16.6.6 SS1 通道的高级编程接口

SynCTI驱动程序实现的呼叫状态机已经能够满足大多数的应用场合。如果应用程序希望自行实现呼叫过程，可以使用低级API函数。配置项EnableAutoCall用于关闭驱动程序实现的状态机，也可以在系统启动后通过调用函数SsmSetAutoCallDirection实现。驱动程序关闭SS1 状态机后，通过收/发CAS、R2 自行实现SS1 协议栈及MFC互控过程。

有关 1 号信令接续控制的事件包括：

- E_RCV_CAS
- E_RCV_R2

有关中国 1 号信令接续控制的高级编程接口函数包括：

- SsmSendCAS
- SsmGetSendingCAS
- SsmGetCAS
- SsmSetRxR2Mode
- SsmGetR2
- SsmSendR2
- SsmSendR2Ex
- SsmStopSendR2
- SsmGetSendingR2
- SsmEnableRxDtmf

注意：

- **应用程序使用底层接续控制函数完成接续过程后，如果希望接收 DTMF 按键字符，则必须调用函数 SsmEnableRxDtmf 以开启 DTMF 接收器。同样，通话结束后也应该调用该函数以关闭 DTMF 接收器。**
- **如果希望应用程序能够无缝地从 SS1 信令移植到 SS7 信令，则不应该使用本小节的函数来自行控制呼叫接续过程，而应该使用系统提供的自动呼叫接续函数**

1.16.6.7 No.1 信令的配置

使用 SS1 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	***PcmNumber***	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	PcmSSx	
	PcmClockMode	
	PcmLinkType	
	EnableAutoCall	SS1 信令自动接续控制
	AutoCallInTimeSlot	SS1 信令通道出中继，入中继方向设置
[PcmInfo]	***TotalPcm***	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[SS1Config]	TxCas_CD	SS1 信令 CAS、R2 接续参数设置
	RxCASFilterTime	
	MaxWaitMfcTime	
	IsBlockSS1In	
	ProtocolType	Lineside 协议类型选择及通道状态迁移条件参数设置
	LSWaitPickup	
	Ss1SendIdleState	
	mfcr2_Protocol	MFCR2 协议国家选择以及相应参数设置
	MaxWaitAutoDialAnswerTime	SS1 去话呼叫：设置工作参数
	MaxWaitOccupyAckTime	
	MaxWaitKBTime	
	RepeatPhoNumOn1stR2bwdIsA5	
	MfcKD	SS1 来话呼叫：设置工作参数
	MfcKA	
	TxCallerId	
	MaxWaitSetKBTime	
	MaxWaitKDTime	
	ToRingingDelayTime	
	A1ToA3pWaitTime	
	MfcKB	
	AutoSendKB	
	PhoNumHoldup	
	A3pTime	
	DefaultRcvPhoNumLen	
	DefaultRcvCallerID	
	MaxPhoNumRule	
	Rule	
	Ss1OutputLog	SS1 信令 CAS、R2 日志输出设置
	MfcR2ToRxCallerIdBuf	
	IsBlockSS1In	

注：***XXXX*** 字样为必须配置的项目，XXXX 为可选的配置项目

1.16.7 连接到远端坐席接入模块

SHD 系列板卡可以直接连接到远端坐席接入模块（Channel bank），以构建大容量的坐席系统。在连接到远端坐席接入模块后，SHD 系列板卡的通道类型将变成坐席通道，具有 SHT 系列板卡中坐席通道的全部属性。

配置项UsageMode用于决定是将SHD系列板卡作为呼叫用板卡，还是连接到远端坐席接入模块。

配置项CBProtocolType用于设置连接到远端坐席接入模块时使用的信令方式。

配置项CBChannelType用于设置连接到远端坐席接入模块时的通道类型。

SHD 系列板卡连接到远端坐席接入模块后，通道的外部特性与对应的 SHT 系列板卡通道的外部特性完全相同，因此，通道的工作参数的配置项和配置方法也完全一样，详细信息请参见 SHT 板卡的相关内容。

1.17 SHV 系列板卡（CTI 系列）

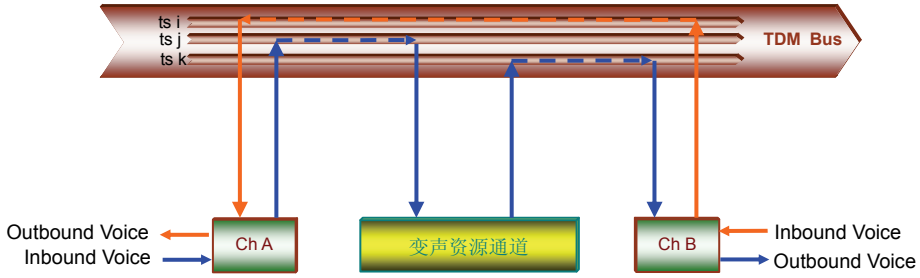
1.17.1 SHV 系列板卡简介

SHV 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Max. Ports	TDM Bus
SHV-120A-CT/PCI	PCI	120	H.100
SHV-240A-CT/PCI	PCI	240	H.100
SHV-240A-CT/cPCI	cPCI	240	H.110

1.17.2 SHV 系列板卡的典型应用原理框图

SHV 系列板卡的典型应用如下图所示：



上图中，chA 与 ChB 之间通过 TDM 交换总线建立双向连接，ChB 的来话信号经时隙 i 连接到 ChA，形成 ChA 的去话信号；ChA 的来话信号经时隙 j 进入变声资源通道，ChA 的来话信号在变声资源通道内进行变声处理后，通过时隙 k 连接到 ChB，形成 ChB 的去话信号。这样，ChA 听到 ChB 的声音为原始声音，而 ChB 听到 ChA 的声音为变声后的声音。

1.17.3 有关变声功能的函数一览表

SHV 系列板卡是资源卡，必须与 SHD 系列或 SHT 系列配合使用。有关变声功能的函数如下表所示：

函数名称	描述
SsmGetMaxVCh	获取应用系统内变声资源通道的总数
SsmGetMaxFreeVCh	获取空闲的变声资源通道的总数
SsmBindVCh	将语音通道绑定到变声资源通道
SsmUnbindVCh	解除语音通道与变声资源通道的绑定
SsmSetVoiceEffect	设置变声的效果
SsmGetVoiceEffect	获取变声的效果

1.18 SHN 系列板卡（CTI 系列）

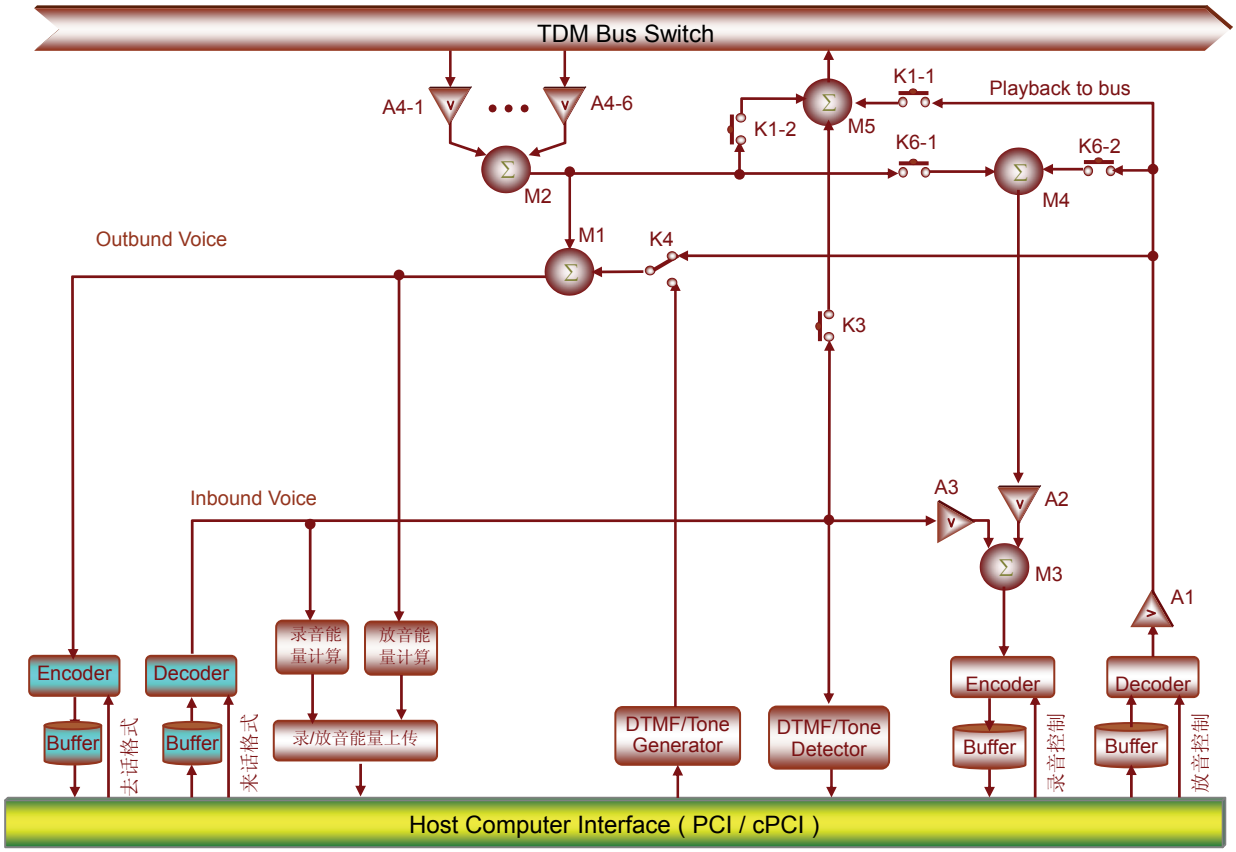
1.18.1 SHN 系列板卡简介

SHN 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	Voltage Detector	TDM Bus	Max Fax Resource	SIP	H.323
SHN-32A-CT/PCI	PCI	32	√	√	√	√	N/A	√	√

1.18.2 SHN 系列板卡的工作原理框图

SHN 系列板卡的工作原理图如下图所示：



原理框图中有关录音操作的符号说明：

名称	描述
M1	去话混音器
M2	下总线信号混音器
M3	录音混音器。 M3 的输出进入编码器（Encoder）模块，然后形成录音数据。
M4	录音用途的去话混音器。 用于将下总线信号混音器的输出、放音数据、信号音发生器/DTMF 发生器的输出进行混音，混音结果进入录音混音器 M3。
M5	上总线混音器。 用于将放音数据、下总线信号混音器 M2 的输出、来话信号进行混音，混音结果上行到 TDM 交换总线。板卡初始化时，会将M5 的输出上行到TDM总线的某个时隙（记为ts）上，函数SsmLinkToBus可以用来改变ts。
A1	放音的音量调节器，可以通过配置项DefaultPlayVolume设置，也可通过函数SsmSetPlayVolume或SsmSetPlayGain进行调节，缺省的音量设置为 0。
A2	去话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecMixer进行设置，也可以通过配置项DefaultRecordMixerVolume设置，缺省值-7，关闭A2。
A3	来话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecVolume进行设置，也可以通过配置项DefaultRecordVolume设置，缺省值为 0。

A4-1 … A4-6	下总线信号进入M2 前的音量调节器，可以通过函数SsmSetListenVlmInConf进行设置，缺省值为 0DB。
K1-1	控制放音数据是否进入混音器M5 并进而上行到TDM交换总线上的开关，通常为断开状态，可以通过函数SsmSetPlayDest进行控制。K1-1 仅用于电话会议中给会议室播放背景音。
K1-2	M2 的输出信号是否重新回到TDM 交换总线的控制开关，可以通过函数SsmSetFlag（携带F_MixerResToBus参数）进行设置，缺省值为断开。
K3	来话信号是否进入上总线混音器M5 的控制开关，可通过函数SsmSetFlag（携带F_InVoiceToBus参数）或配置项InVoiceToBus进行设置，缺省值为闭合。
K4	发送 DTMF/信号音与去话方向放音不能同时进行。
K6-1 K6-2	开关K6-1 控制总线混音器M2 的输出是否作为录音混音器M3 的一个输入信号源；开关K6-2 控制放音数据是否作为录音混音器M3 的一个输入信号源。函数SsmSetRecBack用来设置这 2 个开关。

➢ **放音操作**

应用程序调用播放函数时，解码后的语音数据经过音量调节器 A1 进行音量调节后，可以有 3 个去向：

- ✧ 进入 M1 混音器，与其它信号源混音后形成去话信号；
- ✧ 在 K6-2 控制下，进入 M4 混音器，与其它信号源混音后形成录音操作的信号源；
- ✧ 在 K1-1 控制下，进入混音器 M5，与其它信号源混音后形成上总线的信号源，仅用于电话会议中给会议室播放背景音。

➢ **录音操作的信号源选择范例**

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

✧ 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);          // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);    // 关闭 A2
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

✧ 范例 2：ch1 与 ch2 建立双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）和总线混音器 M2 的输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);            // 建立双向连接: ch1↔ch2
SsmSetRecBack(ch1,2);             // K6-1 闭合，K6-2 断开
SsmSetRecVolume(ch1, 2);          // 开启 A3，设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);    // 开启 A2，设置增益为-1*3DB=-3DB
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

✧ 范例 3：ch1 以主持人身份加入了一个电话会议室（假设会议室编号为 n），在 ch1 上播放背景音乐，同时通过 ch1 对电话会议进行录音（包括背景音乐）。所有信号源均为正常音量，即增益为 0。实现代码为：

```
SsmJoinConfGroup(n, ch1, ……);     // ch1 加入会议室：K3 闭合
SsmSetPlayDest(int ch, 1);        // 闭合 K1-1
SsmSetRecBack(ch1,1);             // K6-1 闭合，K6-2 闭合
SsmPlayFile(ch1, ……);              // ch1 启动放音任务：K4 连接左右两端
SsmSetRecVolume(ch1, 0);          // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, TRUE, 0);     // 开启 A2，设置增益为 0DB
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

1.18.3 SIP 通道的编程

1.18.3.1 SIP 编程基本概念

➢ **SIP 消息**

有 2 种类型的 SIP 消息：

- Requests（请求消息）：从 Client 发送给 Server 的请求消息
- Responses（响应代码）：从 Server 发送给 Client 的响应消息

下表是 SHN 系列板卡 SIP 通道支持的请求消息列表：

Method	Description
INVITE	发起呼叫、修改呼叫参数(re-INVITE)
ACK	对 INVITE 最终响应的确认
BYE	结束一个会话
CANCEL	取消正在发起的呼叫
OPTIONS	查询远端的能力
REGISTER	注册
INFO	发送信息，在 SHN 系列板卡中也用于发送带外 DTMF 字符
MESSAGE	传输即时数据
SUBSCRIBE	订阅用户代理的状态
NOTIFY	通知订阅者当前状态

➢ **消息分段**

SIP 消息由以下几个部分组成：

- 一个起始行
- 一个或多个头字段
- 一个标识头字段结束的空行
- 一个可选消息体

起始行、每个消息头行和空行必须以回车换行序列（CRLF）终止。注意：即使没有消息体，也必须有空行。

➢ **SDP**

SDP 是一个用来描述会话的协议，SIP 消息携带 SDP 的方式类似于 EMAIL 中携带文挡或是 HTTP 中携带 WEB 页面。该协议定义在 RFC2327 中，它规定了对描述会话的必要信息如何进行编码。

SDP 包通常包括：会话信息（Session information）和媒体信息（Media information）。

会话信息包括：

- 会话的主人和会话标识
- 会话激活的时间

考虑到会话的资源受到会话的各方限制，因此还可以包括下面这些实用信息：

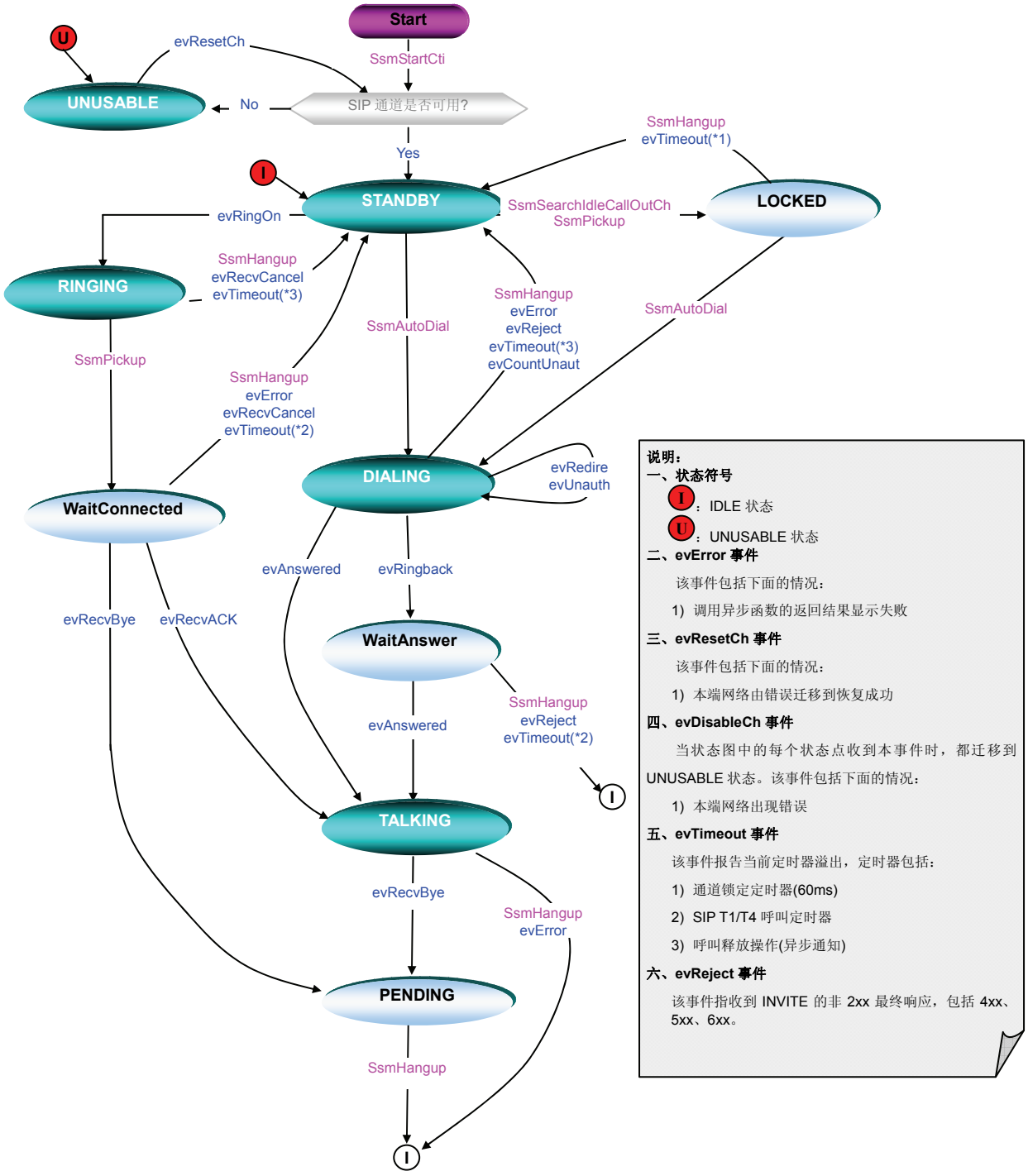
- 带宽信息
- 会话的联系信息，如：E-Mail 地址、电话号码

媒体信息包括：

- 媒体的类型，如：video、audio

- 传输的协议，如：RTP/UDP/IP、H.320
- 媒体的格式，如：H.261 video and MPEG video
- 多播的地址和传输的端口 （IP multicast session）
- 联系信息中的远程地址和传输端口（IP unicast session）

1.18.3.2 SIP 通道的状态转移图



各状态标识的说明如下表所示：

状态标识	描述
Unusable	通道“不可用”状态。 ✧ 在状态图中的任何状态点上收到 evDisableCh 事件，通道迁移到不可用状态，输出[Unusable]。
Standby	“空闲”状态，板卡准备就绪后，进入空闲状态。 ✧ 当调用函数SsmAutoDial发起一个呼叫时，迁移到[DIALING]状态。 ✧ 当调用函数SsmSearchIdleCallOutCh查找可用通道时，迁移到[Locked]状态。 ✧ 当调用函数SsmPickup时，锁定一个空闲通道，并迁移到[Locked]状态。 ✧ 当前接收到呼入并有空闲通道时，迁移到[Ringing]状态。 ✧ 当前接收到呼入但是没有空闲通道时，A3 自动发送 403 响应码并设置原因。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
DIALING	“等待呼叫结果”状态，在收到远端的响应前将一直保持在这个状态。 ✧ 在该状态下，如果收到 evRingback 事件(收到远端发送 180 响应)，状态迁移到[WaitAnswer]。 ✧ 如果收到 evRedire 事件(收到远端发送 3xx 响应)，SynSIP 依靠该通道进行再次呼叫，并自动发送 ACK 消息到远端。 ✧ 如果收到 evUnauth 事件(收到远端发送 401/407 响应)，SynSIP 自动发送带认证信息的呼叫请求到远端。 ✧ 如果收到 evAnswered 事件(收到远端发送 2xx 响应)，状态迁移到[Talking]，同时 SynSIP 会自动发送 ACK 消息到远端。 ✧ 如果收到 evReject(对端拒绝呼叫)、evError(拨号为异步函数)、evTimeout(定时器溢出)、evCountUnauth(认证未能通过)，状态迁移到[Standby]，同时设置原因码。此时 SynSIP 会依据当前事件(evErrAnsw 和 evCountUnauth)的需要决定是否发送 ACK 消息到远端。 ✧ 如果用户在该状态下调用函数SsmHangup挂机，状态迁移到[Standby]，同时SynSIP会自动发送CANCEL消息到远端。 ✧ 通道进入[Dialing]状态时，SynSIP 会启动定时器 B、C。如果应用程序在定时器溢出之前没有发生状态迁移，驱动程序会对此通道进行回收，将通道重新迁移到空闲状态。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
Locked	“呼出锁定”状态。 ✧ 应用程序调用SsmSearchIdleCallOutCh函数时，驱动程序会将函数返回的通道临时锁定，同时启动定时器T300。如果应用程序在T300 溢出之前没有调用SsmPickup或SsmAutoDial函数，驱动程序就会对此通道进行回收。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
WaitAnswer	“回铃”状态。 ✧ 在该状态下，如果收到 evAnswered 事件(收到远端发送 2xx 响应)，状态迁移到[Talking]。 ✧ 如果收到 evReject(对端拒绝呼叫)、evTimeout(定时器溢出)，状态迁移到[Standby]，同时设置原因码。此时 SynSIP 会依据当前错误原因决定是否发送 ACK 消息到远端。 ✧ 用户如果在该状态下调用函数SsmHangup挂机，状态迁移到[Standby]，同时SIP栈会自动发送CANCEL消息到远端。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
Ringing	被叫“振铃”状态。当接收到呼入请求并且有空闲通道时，SynSIP 自动回应 180 响应，锁住当前通道，并迁移到本状态。 ✧ 在该状态下，用户决定接受当前呼叫请求，调用函数SsmPickup，状态迁移到[WaitConnected]。此时SynSIP自动产生 200 OK响应发送到对端。 ✧ 用户也可以拒绝呼叫，此时调用函数SsmHangup拒绝呼叫，状态迁移到[Standby]。此时SynSIP自动产生 403 Forbidden响应发送到对端。 ✧ 如果收到 evRecvCancel 事件(主叫取消呼叫)，状态迁移到[Standby]。此时 SynSIP 自动产生 200 OK 响应 CANCEL 消息。 ✧ 如果收到 evTimeout（定时器溢出），状态迁移到[Standby]。SynSIP 自动产生发送 400 错误响应。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
WaitConnected	“等待连接”状态。 ✧ 在该状态下，用户如果调用函数SsmHangup挂机，状态迁移到[Standby]。此时SynSIP自动产生BYE消息发送到主叫。 ✧ 如果收到 evRecvCancel 事件(主叫取消呼叫)，状态迁移到[Standby]，SynSIP 产生 200 OK 响应相关 CANCEL 消息。 ✧ 如果收到 evRecvAck 事件(主叫已连接)，状态迁移到[Talking]。 ✧ 如果收到 evRecvBye 事件，状态迁移到[Pending]。SynSIP 产生 200 OK 响应 BYE 消息。

	✧ 如果收到 evError、evTimeout(定时器溢出)，状态迁移到[Standby]。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
Talking	通话状态。此状态下，音频通道都已连接。 ✧ 在此状态下，主被叫都可通过函数SsmHangup挂机，此时SynSIP自动产生BYE消息通知对方，同时状态迁移到[Standby]。 ✧ 如果收到 evRecvBye 事件(对方挂机)，状态迁移到[Pending]。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。
Pending	“挂起”状态。 ✧ 在呼叫状态机迁移的过程中，呼叫接收到SIP协议规定的被终止状态，此时呼叫已无法建立，用户必须在此释放呼叫过程中分配的所有资源及复位电路。必须调用函数SsmHangup，线路才会复位到[Standby]状态。 ✧ 收到 evDisableCh（网络错误、无可用通道），迁移到[Unusable]状态。

各状态标识在头文件 ShpA3Api.h 中的声明如下表所示：

状态标识	ShpA3Api.h 中的值	ShpA3Api.h 中的宏定义
Unusable	134	S_CALL_VOIP_CHANNEL_UNUSABLE
Standby	0	S_CALL_STANDBY
Dialing	132	S_CALL_VOIP_DIALING
Locked	12	S_CALL_LOCKED
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Ringing	2	S_CALL_RINGING
WaitConnect	133	S_CALL_VOIP_WAIT_CONNECTED
Talking	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING

驱动程序内部逻辑判断模块的描述如下表所示：

模块输出	判决条件
Unusable	若事件 evDisableCh、DataLinkDisconnect、Circuit Reset 之一保持有效，则通道不可用。
Standby	若 evDisableCh 事件保持无效，DataLinkConnect 事件保持有效，且电路重起成功，则通道状态为空闲。

注意：

在此模块中，驱动程序还要做下列清理工作：

- 清零并关闭信号音检测器；
- 如果应用程序启动了 WaitDtmf 任务，该任务将被自动终止；
- 如果应用程序启动了 DTMF 发送任务，该任务将被自动终止；
- DTMF检测器的缓冲区将被自动清空。如果配置项AlwaysEnableRxDtmf设置为 0， DTMF检测器将被自动关闭；
- 如果配置项AlwaysDetectBargeIn设置为 0，Barge-in 检测器将被自动关闭；
- 如果配置项AutoClearCallerIdBufOnHangup设置为 1，存放主叫号码的缓冲区将被自动清空；
- 所有 SynSIP 中的定时器复位。

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_PROC_AutoDial	应用程序调用SsmAutoDial函数即启动了AutoDial任务。当AutoDial任务有进展时，驱动程序向应用程序抛出本事件。

注：事件的声明在文件 ShpA3Api.h 中。

1.18.3.3 SIP 通道的配置

使用 SIP 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	*ProtocolType*	设置板卡上使用的 IP 通信协议类型，以及接收 DTMF 的方式
	RecvDtmfType	
[SIP]	*LocalIp*	设置 SIP 协议的侦听地址和端口
	LocalPort	
	RTPRange	设置 SIP 会话时的工作参数，依次是： ✧ RTP 端口范围 ✧ 显示名称 ✧ 发送 DTMF 的方式 ✧ 使用的语音编解码
	OutboundSrv	
	DisplayName	
	SendDtmfType	
	AudioCodecList	
	UserName	设置与注册相关的参数，依次是： ✧ 当启动注册时，即注册用户名；否则设置为 SIP Uri 中@前的用户名 ✧ 注册服务器 ✧ 注册时使用的密码（如果需要） ✧ 注册有效时间，默认为 3600
	Domain	
	RegPassword	
	RegExpires	
	LogFile	设置板卡高级配置项，依次是： ✧ 日志保存的文件 ✧ 日志等级 ✧ 端口保持心跳间隔（用于 NAT 穿透）
	LogLevel	
	HeartInterval	

注：*XXXX* 字样为必须配置的项目，XXXX 为可选的配置项目

1.18.4 H.323 通道的编程

1.18.4.1 H.323 编程基本概念

1.18.4.1.1 H.323 实体

H.323 的基本组成单元是“域”。在 H.323 系统中，所谓“域”是指一个由网守（Gatekeeper）管理的网关（Gateway）、多点控制单元（MCU）、多点控制器（MC）、多点处理器（MP）和所有终端组成的集合。H.323 系统

中各个逻辑组成部份称为 H.323 的实体，包括以下这些：

- 终端（H.323 Endpoint）
- 网关（Gateway）
- 网守（Gatekeeper）
- 多点控制单元（MCU）
- 多点控制器（MC）
- 多点处理器（MP）

H.323 网守可以是软件或硬件设备。其主要功能有：地址转换、许可控制、带宽管理、区域管理、呼叫控制信令、呼叫鉴权和呼叫管理。前四个为基本功能。在 H.323 的建议中，网守是一个可选部件，但是对于运行在公网上的 IP 电话系统来说，网守是一个必须的部件。

1.18.4.1.2H.323 消息

H.323 协议中主要有两类消息：H.225.0 和 H.245 消息。

H.225.0 又分为 RAS 信令功能和呼叫信令功能两个部分。RAS 信令功能就是采用 H.225.0 消息在终端和网守之间实现注册、接入许可、带宽修改、状态查询和拆除程序。呼叫信令功能包括呼叫的建立和拆除等流程，它与 Q.931 类似。

H.245 是多媒体通信控制协议，它的控制功能是采用 H.245 控制通路交换端到端控制信息，包括能力交换、主从确定和打开/关闭逻辑通路等。

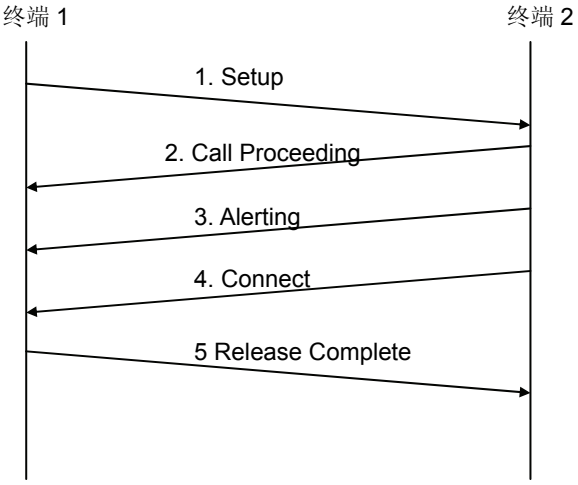
RAS、呼叫信令通路和 H.245 控制通路是各自独立的，没有网守的网络环境是不使用 RAS 信令的。

详细情况请参考 ITU 的 H.323 协议栈定义。

1.18.4.1.3实体交互

H.323 实体间的通讯主要分为有网守和没有网守两种情况。有网守的时候又分为终端直接选路和网守选路，以及单网守和多网守等多种情况。下面主要对三汇 H.323 板卡最常用的没有网守和两个终端在一个网守上直接选路的 2 种模型做简单介绍。

1.18.4.1.3.1 终端直接呼叫，没有网守



1、终端 1 发送 Setup 消息，发起呼叫；

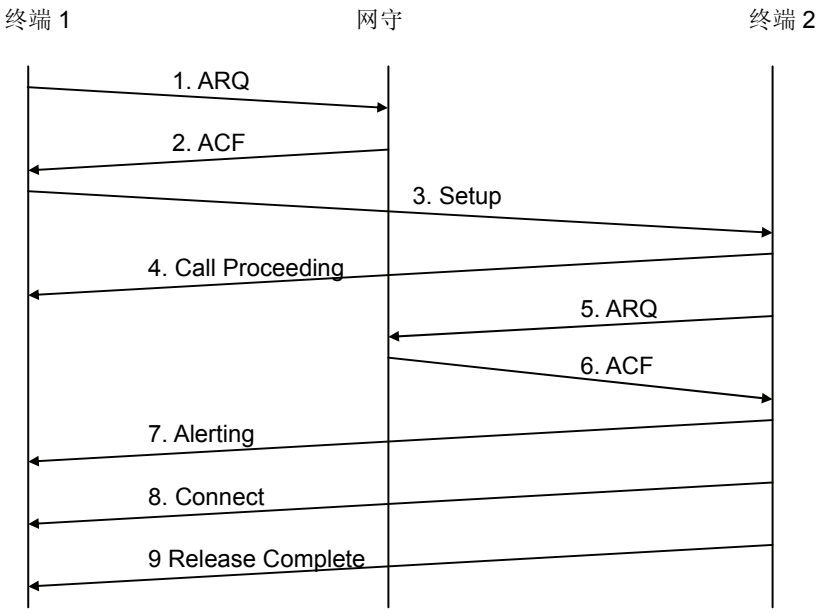
2、终端 2 回送 Call Proceeding 消息，表示呼叫已经到达，正在处理中；

3、终端 2 回送 Alerting 消息，表示振铃中；

4、终端 2 用户应答，向终端 1 发送 Connect 消息，建立媒体通道；

5、终端 1 发送 Release Complete 消息请求结束呼叫。

1.18.4.1.3.2 两个终端在同一个网守上注册



1、终端 1 在 RAS 信道上向其网守发送 ARQ 消息，请求发起至终端 2 的呼叫；

2、网守同意接纳此呼叫，并翻译得出终端 2 的呼叫信令信道运输层地址，由 ACF 消息返回到终端 1；

3、终端 1 发送 Setup 消息发起呼叫；

4、终端 2 回送 Call Proceeding 消息，表示呼叫已经到达，正在处理中；

5、终端 2 愿意接收此呼叫，经 RAS 信道向网守发送 ARQ 消息，请求接收此呼叫；

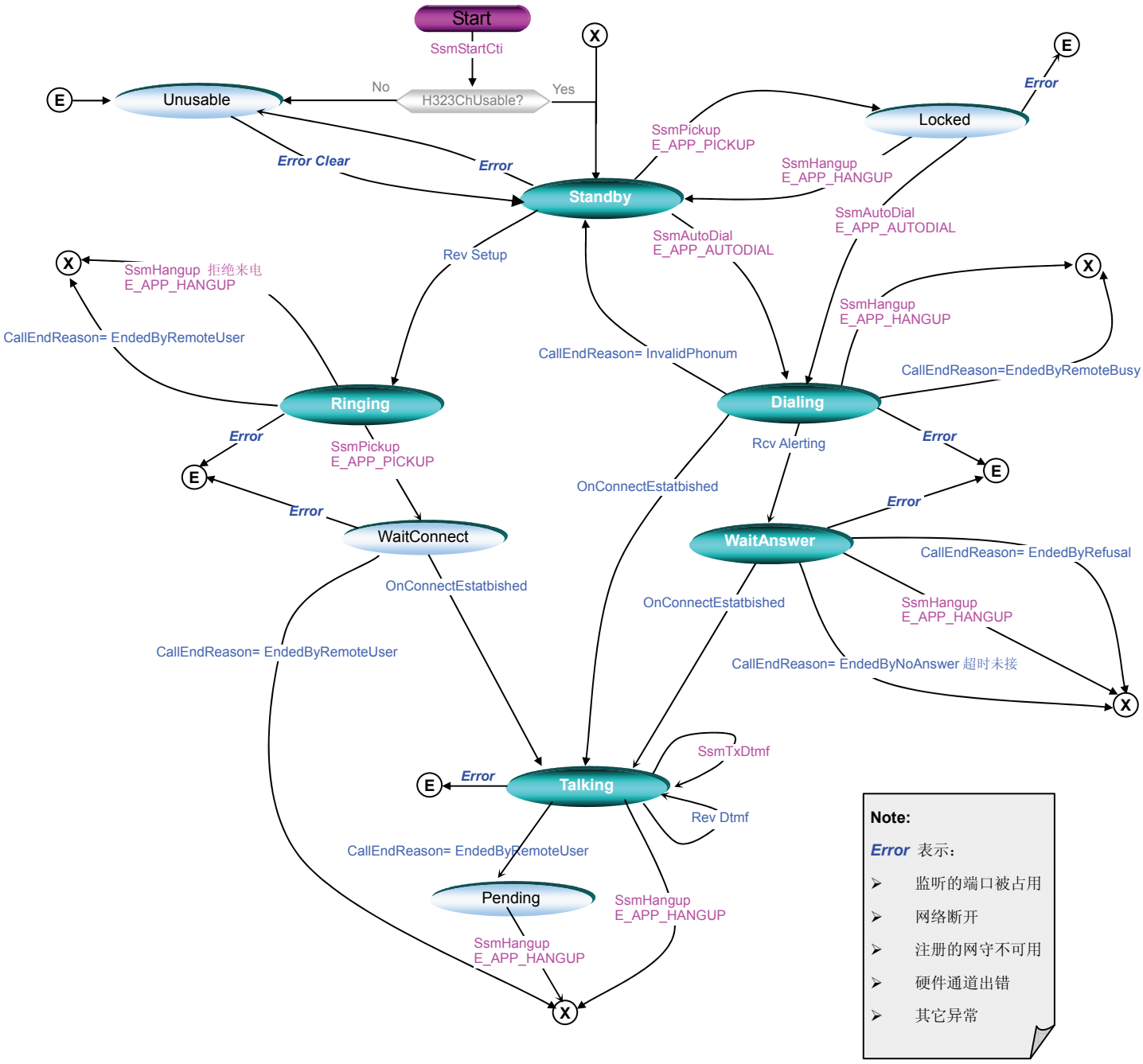
6、网守同意接纳，回送 ACF；

7、终端 2 回送 Alerting 消息，表示振铃中；

8、终端 2 用户应答，向终端 1 发送 Connect 消息，建立媒体通道；

9、终端 2 发送 Release Complete 消息请求结束呼叫。

1.18.4.2 H.323 通道的状态转移图



1.18.4.3 H.323 通道的配置

使用 H.323 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	*ProtocolType*	设置板卡上使用的 IP 通信协议类型，以及接收 DTMF 的方式
	RecvDtmfType	
[H.323]	*NetworkType*	设置 H.323 协议承载的网络协议类型
	LocalIp	设置 H.323 协议的侦听地址和端口
	LocalPort	
	GlobalName	设置 H.323 会话时的工作参数，依次是： • 全局名称/用户名 • 快速启动 • 隧道机制 • Jitter 时延 • RTP 端口范围 • 发送 DTMF 的方式 • 使用的语音编解码
	FastStart	
	Tunnel	
	JitterDelay	
	RTPRange	
	SendDtmfType	
	AudioCodecList	
	GateKeeper	设置与注册相关的参数，依次是： • 网守地址 • 用户密码 • 注册终端类型
	GKPassword	
	EndPointType	
	LogLevel	设置板卡高级配置项： • 日志等级

1.19 DST 系列板卡（REC 系列）

1.19.1 DST 系列板卡简介

数字电话交换机，如 Siemens HIPath、Alcatel 4200、Alcatel 4400 系列等，与数字电话机相连时，电话线上传输的不是模拟音频信号，而是纯数字信号。数字电话交换机与数字电话机之间的物理连接线被称为数字用户线。数字用户线由一个的 D 信道和 2 个 B 信道组成，每个信道的速率为 64kbps。 D 信道用于传输与呼叫相关的信令信息，2 个 B 信道可以同时用来传输语音数据和其它数据，如传真数据等。

DST 系列板卡是用来采集数字用户线上 B 信道的语音数据和 D 信道的信令数据的专用录音卡。DST 系列板卡

的线路输入部分采用了高阻电路，信号采集电路使用了现场可编程的芯片，通过应用程序的简单设置，就可以使同片板卡支持不同型号的交换机和数字电话机。

DST 系列板卡的语音处理能力包括：

- 采用高阻方式采集 B 信道上的语音数据，不对原有通话双方造成任何干扰，也不会被原有通话双方所察觉；
- 以指定的语音编码格式对语音数据进行压缩和录制；
- 录音数据的自动增益控制（AGC）
- 检测 DTMF 信号；
- 检测语音活动；
- 检测特定频率的单音频和双音频信号音；
- 实时播放采集的语音信号；
- 总线交换；
- 回放采集的语音数据；

D 信道上的信令信息可以用来获取电话呼叫的相关信息。DST 系列板卡的 D 信道信令处理能力包括检测数字电话机的下列操作或状态变化：

- 摘机(Off-hook)、挂机(On-hook)；
- 拨号键(Dial keys)；
- 功能键(Function keys)；
- 语音通道的开/关（Voice Channel open/close）；
- 指示灯(LED)；
- 屏幕(LCD)上的显示内容

DST 系列板卡的型号如下表所示：

Board Model	Form Factor	Max. Ports per Board	Module Type	Max. Modules per Board
SHR-16DA-CT/PCI	PCI / PCI-X	16	MOD_16DA	4
SHR-24DA-CT/PCI	PCI / PCI-X	24	MOD_24DA	3
SHR-2DA/USB/D	USB	2	—	—

DST 系列板卡支持的模块型号如下表所示：

Module Type	2 线输入时的通道数	4 线输入时的通道数
MOD_16DA	4	2
MOD_24DA	8	4

1.19.2 DST 系列板卡支持的交换机型号

DST 系列板卡支持的交换机型号如下表所示：

生产厂家	交换机型号	话机型号	线路接口	要求的 SynCTI 版本		相关配置项的设置值	
				DST 状态机	输出 D 信道原始消息	PBXType	PhoneType
Siemens	Hicom 100E Hicom 150E	Optiset E** advance	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		Optiset E** advance conf	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		Optiset E basic	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		Optiset E** standard	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0

		AC 3.0	2	4.6.0.6		Siemens Hicom/Hipath	1
		AC 4.0	2	4.6.0.6		Siemens Hicom/Hipath	1
		AC WIN 3.0	2	4.6.0.6		Siemens Hicom/Hipath	2
		AC WIN 4.0	2	4.6.0.6		Siemens Hicom/Hipath	2
	Hicom 300E HiPath 3000 series HiPath 4000 series Hipath 3550	Optiset E Basic	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		Optiset E Standard	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		Optiset E Advanced Conf	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		OptiPoint 500 advance	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		OptiPoint 500 basic	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		OptiPoint 500 entry	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		OptiPoint 500 standard	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		OptiPoint 500 standard SL	2	4.5.6.3	4.7.3.1	Siemens Hicom/Hipath	0
		AC 3.0	2	4.6.0.6		Siemens Hicom/Hipath	1
		AC 4.0	2	4.6.0.6		Siemens Hicom/Hipath	1
		AC WIN 3.0	2	4.6.0.6		Siemens Hicom/Hipath	2
		AC WIN 4.0	2	4.6.0.6		Siemens Hicom/Hipath	2
Alcatel	4200	4004 Reflex first	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
		4010 Reflex easy	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
		4020 Reflex premium	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
		4035 Reflex advanced	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
	OmniPCX 4400	4004 Reflex first	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
		4010 Reflex easy	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0

		4020 Reflex premium	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
		4035 Reflex advanced	2	4.5.6.0	4.7.3.1	Alcatel 4200/4400	0
		4049	2	4.7.1.7		Alcatel 4200/4400	3
		4059	2	4.7.1.7		Alcatel 4200/4400	2
	OmniPCX Enterprise(OXE)	4029	2	4.7.1.7		Alcatel 4200/4400	1
		4011	2	4.7.1.8		Alcatel 4200/4400	4
	4300	4304	2	4.7.1.7		ALCATEL4300	0
		4321	2	4.7.1.7		ALCATEL4300	0
Nortel	Meridian M1SL100 CS1000 C11	2006	2	4.7.3.1	4.7.3.1	Nortel Meridian1	0
		2008	2	4.5.7.2	4.7.3.1	Nortel Meridian1	0
		2216	2	4.5.7.2	4.7.3.1	Nortel Meridian1	0
		2616	2	4.7.1.3	4.7.3.1	Nortel Meridian1	0
		*3820	2	4.7.3.1	4.7.3.1	Nortel Meridian1	0
		3901	2	4.7.3.1	4.7.3.1	Nortel Meridian1	0
		3902	2	4.5.7.2	4.7.3.1	Nortel Meridian1	0
		3903	2	4.7.3.1	4.7.3.1	Nortel Meridian1	0
		3904	2	4.7.3.1	4.7.3.1	Nortel Meridian1	0
		3905	2	4.5.7.2	4.7.3.1	Nortel Meridian1	0
		2250	4	4.7.1.2		Nortel Meridian1	1
	NORSTAR	7100	2	4.7.3.1	4.7.3.1	NORSTAR	0
		7208	2	4.7.3.1	4.7.3.1	NORSTAR	0
		7316	2	4.7.3.1	4.7.3.1	NORSTAR	0
		7310	2	4.7.3.1	4.7.3.1	NORSTAR	0
		7324	2	4.7.3.1	4.7.3.1	NORSTAR	0
	Norstar BCM	7100	2	4.7.3.1	4.7.3.1	NORSTAR	1
		7208	2	4.7.3.1	4.7.3.1	NORSTAR	1

		7316	2	4.7.3.1	4.7.3.1	NORSTAR	1
		7310	2	4.7.3.1	4.7.3.1	NORSTAR	1
		7324	2	4.7.3.1	4.7.3.1	NORSTAR	1
Avaya	DEFINITY 2 wire	6402	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		6408	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		8410	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		8434	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		6402 display	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		6408 display	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		6416 display	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		6424 display	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		Call Master III (603E)	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		Call Master IV (603H)	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		Call Master V (607A)	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
		Call Master VI (606A)	2	4.5.8.0	4.7.3.1	Avaya Definity 2W	0
	S8300 2 wire	2410	2	4.7.3.1	4.7.3.1	Avaya Definity 2W	0
		2420	2	4.7.3.1	4.7.3.1	Avaya Definity 2W	0
	IP Office	2410	2	4.7.3.1	4.7.3.1	Avaya Definity 2W	0
		2420	2	4.7.3.1	4.7.3.1	Avaya Definity 2W	0
		5410	2	4.7.3.1	4.7.3.1	Avaya Definity 2W	0
		5420	2	4.7.3.1	4.7.3.1	Avaya Definity 2W	0
	DEFINITY 4 wire	7406	4	4.7.3.1	4.7.3.1	Avaya Definity 4W	0s
		8410	4	4.7.3.1	4.7.3.1	Avaya Definity 4W	0

		Call Master II (603A)	4	4.7.3.1	4.7.3.1	Avaya Definity 4W	0
		Call Master III (603E)	4	4.7.3.1	4.7.3.1	Avaya Definity 4W	0
		Call Master IV (603F)	4	4.7.3.1	4.7.3.1	Avaya Definity 4W	0
		CallMaster I (602A)	4	4.7.3.1	4.7.3.1	Avaya Definity 4W	0
	G3Si	6408B+	2	4.8.0.0	4.8.0.0	Avaya Definity 2W	0
	S8500 S8300	6408D+	2	4.8.0.0	4.8.0.0	Avaya Definity 2W	0
		Call Master V	2	4.8.0.0	4.8.0.0	Avaya Definity 2W	0
	8710	2402	2	4.8.0.0	4.8.0.0	Avaya Definity 2W	0
		Call Master V	2	4.8.0.0	4.8.0.0	Avaya Definity 2W	0
NEC	Electra Elite 192	DTU-16D Elite	2	4.7.3.1	4.7.3.1	NEC	0
		DTU-32D Elite	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-16D Series E (75)	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-32D Series E (75)	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-8D Series E (75)	2	4.7.3.1	4.7.3.1	NEC	0
		DCU 60	2	4.7.3.1	4.7.3.1	NEC	0
	NEAX 2400 IMS NEAX 2400 APX	DTR 8D-1 Dterm Series I	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 8-1 Dterm Series I	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 32D-1 Dterm Series	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 2DT-1 Dterm Series	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 16D-1 Dterm Series	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-32DA-1 (DtermSeriesE)	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-32D-1 (DtermSeriesE)	2	4.7.1.5	4.7.3.1	NEC	1
		DTP-16D-1 (DtermSeriesE)	2	4.7.1.5	4.7.3.1	NEC	1
		DTP 8D (DtermSeriesE)	2	4.7.3.1	4.7.3.1	NEC	0
		Dterm Series 3 (16 button)	2	4.7.3.1	4.7.3.1	NEC	0

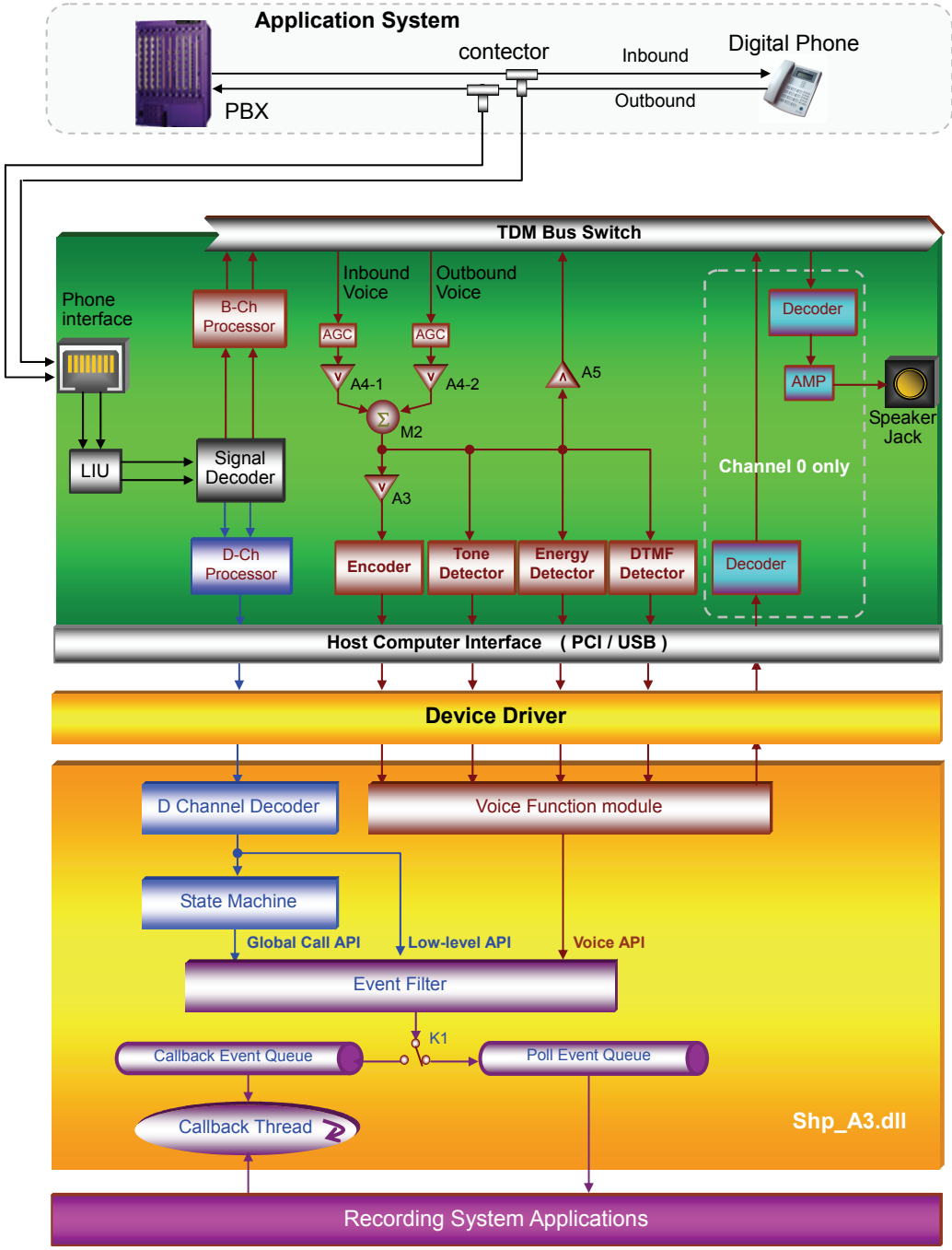
		DCU 60	2	4.7.3.1	4.7.3.1	NEC	0
	EDK21	DTU-32D Elite	2	4.8.0.0	4.8.0.0	NEC	0
		DTP-8D-1	2	4.8.0.0	4.8.0.0	NEC	0
	NEAX 7400	ETJ16DD	2	4.8.0.0	4.8.0.0	NEC	0
	NEAX1000	DTP 8D	2	4.7.3.1	4.7.3.1	NEC	0
		DCU 60	2	4.7.3.1	4.7.3.1	NEC	0
	Professional	DTU-16D Elite	2	4.7.3.1	4.7.3.1	NEC	0
		DTU-32D Elite	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-16D Series E (75)	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-32D Series E (75)	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-8D Series E (75)	2	4.7.3.1	4.7.3.1	NEC	0
		DCU 60	2	4.7.3.1	4.7.3.1	NEC	0
	NEAX 2000	DTR 8D-1 Dterm Series I	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 8-1 Dterm Series I	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 32D-1 Dterm Series I	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 2DT-1 Dterm Series I	2	4.7.3.1	4.7.3.1	NEC	0
		DTR 16D-1 Dterm Series I	2	4.7.1.4	4.7.3.1	NEC	0
		DTP-32DA-1 (DtermSeriesE)	2	4.7.3.1	4.7.3.1	NEC	0
		DTP-32D-1 (DtermSeriesE)	2	4.7.1.5	4.7.3.1	NEC	1
		DTP-16D-1 (DtermSeriesE)	2	4.7.1.5	4.7.3.1	NEC	1
		DTP 8D (DtermSeriesE)	2	4.7.3.1	4.7.3.1	NEC	0
		Dterm Series 3 (16 button)	2	4.7.3.1	4.7.3.1	NEC	0
		DCU 60	2	4.7.3.1	4.7.3.1	NEC	0
	Aspila EX	IP1WW-12TXH	2	4.8.0.0		NEC_AspilaEX	0
		IP1WW-24DTXH	2	4.8.0.0		NEC_AspilaEX	0
Ericsson	Business Phone	Dialog3201 (DBC201)	2	4.7.1.3	4.7.3.1	ERICSSON	0
		Dialog3202 (DBC202)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog3203 (DBC203)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog3212 (DBC212)	2	4.7.1.5	4.7.3.1	ERICSSON	0
		Dialog3213 (DBC213)	2	4.7.1.5	4.7.3.1	ERICSSON	0
		Dialog2601 (DBC601)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog2631 (DBC631)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog2661 (DBC661)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog2662 (DBC662)	2	4.7.3.1	4.7.3.1	ERICSSON	0

	MD110	Dialog3201 (DBC201)	2	4.7.1.3	4.7.3.1	ERICSSON	0
		Dialog3202 (DBC202)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog3203 (DBC203)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog3212 (DBC212)	2	4.7.1.5	4.7.3.1	ERICSSON	0
		Dialog3213 (DBC213)	2	4.7.1.5	4.7.3.1	ERICSSON	0
		Dialog2601 (DBC601)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog2631 (DBC631)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog2661 (DBC661)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog2662 (DBC662)	2	4.7.3.1	4.7.3.1	ERICSSON	0
		Dialog4222	2	4.7.1.3		ERICSSON	1
		Dialog4224	2	4.7.1.5		ERICSSON	4
	MD150	Dialog4223	2	4.7.1.5		ERICSSON	2
		Dialog4224	2	4.7.1.5		ERICSSON	3
Tadicom /Tadiran	Coral	DKT 2320	2	4.5.7.6		TADICOM	0
		DKT 2321	2	4.5.7.6		TADICOM	0
		DKT 2322	2	4.5.7.6		TADICOM	0
		FLEXSET 120 /280 series	2	4.5.7.6		TADICOM	0
	DKT 1110	Digital set	2	4.8.0.0		TADICOM	0
Panasonic	KX-TD88	KXT-7230	2	4.6.0.6		PANASONIC	0
	KX-TDA100	KXT-7625	2	4.7.1.7		PANASONIC	1
		KXT-7630	2	4.7.1.7		PANASONIC	1
		KXT-7665	2	4.7.1.7		PANASONIC	1
		KXT-7433	2	4.7.1.8		PANASONIC	2
	KX-TDA200 KX-TDA600	KXT-7625	2	4.7.1.7		PANASONIC	1
		KXT-7630	2	4.7.1.7		PANASONIC	1
		KXT-7665	2	4.7.1.7		PANASONIC	1
	KX-TP1232	KXT-7433	2	4.7.1.8		PANASONIC	2
eON	eON	6 key	4	4.7.1.4		EON	0
		18 key	4	4.7.1.4		EON	0
		38 key	4	4.7.1.4		EON	0
HARRIS	H-2020(1B+D)	Optic 1 Teleset (1B+D)	2	4.7.1.5		HARRIS	0
		1B+D Console	2	4.7.1.8		HARRIS	1
	H-2020(2B+D)	2B+D Teleset	2	4.7.1.8		HARRIS2B	0
TOSHIBA	DK28	DKT3020C-SD	2	4.7.1.5		TOSHIBA	0
		DKT2020-S	2	4.7.1.5		TOSHIBA	0
		DKT2020-SD	2	4.7.1.5		TOSHIBA	0
	CTX670	DKT3020-S	2	4.7.1.7		TOSHIBA	1
PHILIPS_2	SOPHO iS3030 SOPHO iS3000	ErgoLine D325-2W	2	4.7.1.7		PHILIPS_2	0
PHILIPS_4	SOPHO iS3030 SOPHO iS3000	ErgoLine D325-4W	4	4.7.1.7		PHILIPS_4	0
		ErgoLine D330-4W	4	4.7.1.7		PHILIPS_4	0
IWATSU	ADIX-E	IX-12KTD-2	2	4.7.1.8		IWATSU	0

TECOM	DX1688	DU8812D	2	4.7.1.8		TECOM	0
ISDN	ISDN		4	4.7.3.1		ISDN	0
SAMSUNG	DCS-60i	DCS-S1BW	2	4.7.3.1	4.7.3.1	SAMSUNG_DCS_828	0
		DCS-S2EW	2	4.7.3.1	4.7.3.1	SAMSUNG_DCS_828	0
	DCS-828	DS-24SE-KTS	2	4.7.3.1	4.7.3.1	SAMSUNG_DCS_828	0
	Inforex	DS-24SE-KTS	2	4.7.3.1	4.7.3.1	SAMSUNG_DCS_828	0
LG	Opticon 16	LKD 8DS	2	4.8.0.0	4.8.0.0	TELKOM	0
	LGIPLDK-300BKSU	LDK-30DH	2	4.8.0.0	4.8.0.0	TELKOM	0
Mitel	Sx2000	Superset 4150	2	4.8.0.0	4.8.0.0	Mitel Sx2000	0
		Superset 430	2	4.8.0.0	4.8.0.0	Mitel Sx2000	0
		Superset 401+	2	4.8.0.0	4.8.0.0	Mitel Sx2000	0
		Superset 4001	2	4.8.0.0	4.8.0.0	Mitel Sx2000	0
		Superset 4025	2	4.8.0.0	4.8.0.0	Mitel Sx2000	0
		Superset 420	2	4.8.0.0	4.8.0.0	Mitel Sx2000	0
		Superset 6dn	2	4.8.0.0	4.8.0.0	Mitel Sx2000	1

1.19.3 DST 系列板卡的工作原理框图

DST 系列板卡的工作原理图如下图所示：



原理框图中，各功能模块及符号的含义如下表所示：

名称	描述
M2	下总线信号混音器，用于将去话信号和来话信号进行混音。
A4-1	来话信号（以数字话机侧为基准）的音量调节器，可以通过函数DTRSetMixerVolume进行调节，缺省值为 0。
A4-2	去话信号（以数字话机侧为基准）的音量调节器，可以通过函数DTRSetMixerVolume进行调节，缺省值为 0。
A3	录音音量调节器，可以通过配置项DefaultRecordVolume设置，也可通过函数SsmSetRecVolume进行调节。
A5	用于将混音器的输出信号上行到总线，作为实时监听的信号源，一般情况下不需要操作 A5。

➢ **录音操作的信号源选择范例**

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。

✧ 范例 1：录制来话信号＋去话信号，正常音量。实现代码为：

```
DTRSetMixerVolume(ch, 0, 0, 0);      // 设置 A4-1 增益 0DB，A4-2 增益 0DB
SsmSetRecVolume(ch, 0);              // 开启 A3，设置增益 0DB
SsmRecToFile(ch, ……);                // 启动文件录音任务……
```

✧ 范例 2：只录制来话信号，使用缺省音量。实现代码为：

```
DTRSetMixerVolume(ch, 0, 3, -7);     // 设置 A4-1 增益为 3*3DB=9DB，A4-2 增益为-∞
SsmSetRecVolume(ch, -3);             // 开启 A3，设置增益为-3*3DB=-9DB
SsmRecToFile(ch, ……);                // 启动文件录音任务……
```

➢ **放音操作的数据流向**

只有 0 通道才支持放音操作。当应用程序对 0 通道调用播放函数时，驱动程序自动将放音内容上行到总线，然后再将其下行到板载音频功率放大器。

➢ **板载扬声器输出端口**

每片 DST 系列板卡的 0 通道上都内置了具有模拟音频放大电路的板载扬声器输出端口，可直接将声音信号连接到外置的扬声器或耳机，其主要用途为：

✧ 将本卡上某个通道的来话信号（或其它板卡上某个通道的语音信号经CT-BUS）直接送到扬声器进行播放。函数SsmListenTo或SsmListenToEx用于实现此功能；

✧ 在 0 通道上调用语音播放的函数，将应用程序录制的语音数据直播放到扬声器。有关语音播放函数的详细内容请参见本章中“语音播放”部分。

原理框图中的AMP模块是模拟音频放大电路的功率放大器，它的增益可以通过设函数SsmSetPowerAmpVlm设置。

1.19.4 DST 系列板卡的配置

配置项PBXType和PhoneType分别用来设置交换机和数字电话机的型号；配置项DefaultVoiceFormat用来设置数字用户线上的语音编码格式。

1.19.5 DST 系列板卡的语音操作

DST 系列板卡支持的语音操作包括：

➢ **录音操作**

支持文件录音和内存录音，支持的语音编码格式包括：A-率、U-率、IMA ADPCM、G.729A。更多信息请参见本章中有关“录音操作”部分内容。

➢ **放音操作**

DST系列板卡的 0 通道都支持放音操作，包括播放单个文件和播放单个内存。放音操作支持的语音编码格式包括：A-率、U-率、IMA ADPCM、G.729A。更多信息请参见本章中有关“放音操作”部分内容。

➢ **实时监听**

利用DST系列板卡的板载扬声器输出端口可以实现对任意通道的输入语音信号进行实时监听。应用程序可以调用SsmListenTo、SsmListenToEx、SsmLinkFrom或SsmLinkFromEx启动实时监听，调用SsmStopListenTo或SsmStopLinkFrom停止实时监听。函数SsmSetPowerAmpVlm可以用来调节音量。注意：如果被监听的端口与连接扬声器的端口不在同一片板卡上，则需要使用CT-BUS连接这两片板卡。

- **调节输入语音信号的音量**

函数 DTRSetMixerVolume 可以用来设置交换机至话机或话机至交换机的语音信号的音量。

- **DTMF 检测器**

更多信息请参见本章中有关“DTMF检测器”部分内容。

- **信号音检测器**

更多信息请参见本章中有关“信号音检测器”部分内容。

- **Barge in 检测器**

当无法解析出D信道上的信令信息时，可以利用Barge in检测器的检测结果（检测到语音活动/检测到静默）作为触发条件，来启动或停止录音等操作。更多信息请参见本章中有关“Barge in 检测器”部分内容。

1.19.6 DST 系列板卡的信令编程

D 信道上使用的信令协议没有一个统一的标准，通常由交换机或电话机制造商自行制定，不同的制造商，甚至同一个制造商生产的不同产品系列也会有很大差别。

DST 系列板卡的编程模式有 2 种：

- D 信道事件模式：驱动程序负责提取 D 信道上的原始信令信息，以事件的方式将其提供给应用程序，由应用程序完成呼叫过程的信令分析；
- 通用呼叫状态机模式：驱动程序提取 D 信道上的原始信令信息后，直接送到驱动程序内置的呼叫状态机，由呼叫状态机完成呼叫过程的信令分析。呼叫状态机为各种不同的交换机和话机设计了统一的呼叫流程，因此，应用程序只需要将通道的状态迁移信息作为启动或停止录音操作的依据即可。

D 信道事件模式和状态机模式的特点如下表所示：

编程模式	优点	缺点
D 信道事件模式	可以充分利用 D 信道上的全部原始信令消息来详细记录呼叫过程中的每一个细节和状态，对各种具有不同特性的交换机和话机进行个性化编程，例如电话机的特服功能按键操作、LED 指示灯、LCD 上显示信息等	对于每一种不同型号的交换机和电话机，应用程序都需要为其单独编写的呼叫状态机，代码编写的工作量大，对开发人员的要求比较高
状态机模式	开发人员不必关心具体的交换机和电话机型号，以及相关的呼叫细节，就可以开发出适用于所有交换机和话机型号的通用软件，开发时间短，工作量小，对开发人员的要求比较低	状态机只能提供呼叫过程的状态迁移、被叫方号码等适用于所有交换机和话机的简单信息，但无法获得某种交换机型号或电话机型号所特有的信息

1.19.6.1 使用 D 信道的原始信令消息

D 信道上的原始信令消息中，由交换机发送给话机的消息称为交换机消息，对应的事件称为交换机事件；由话机发送给交换机的消息称为话机消息，对应的事件称为话机事件。常见的交换机事件包括：

- 振铃事件；
- 语音通道事件；
- LCD 显示屏事件；
- 指示灯事件；

常见的话机事件包括：

- 摘机/挂机事件；

- ✧ 按键事件；

另外，当话机上发生了某些特定的操作时，话机不必向交换机报告，因此也不会向交换机发送任何消息。这样，即使驱动程序截获了 D 信道上的全部原始信令消息，也有可能无法得到话机上发生这类操作的任何信息。这类操作包括：

- ✧ 当用户操作话机上的音量调节按键时；
- ✧ 话机在 LCD 显示屏上显示某些不依赖于交换机的信息，如对某些按键的功能进行设置等；
- ✧ 话机改变了语音通道的状态，但不必向交换机报告；
- ✧ 话机上 LED 灯的状态发生改变，但不必向交换机报告。

有关 D 信道原始信令消息的详细说明，请参见《DST 事件参考手册》文档的说明。

应用程序应该根据特定型号交换机/话机的呼叫过程，利用部分或全部 D 信道原始信令消息作为状态迁移的条件，为其建立单独的呼叫状态机，然后根据通道的状态变化，来准确记录呼叫过程中的相关信息，以及相关的语音操作，如启动和停止录音操作等。

对于一些可能影响呼叫状态机的状态迁移、但无法在D信道原始信令消息中获得的事件，如语音信道中的忙音、回铃音等信号音，可以通过驱动程序提供的其它语音处理功能获取，更多信息请参见本章中“语音处理”部分内容。

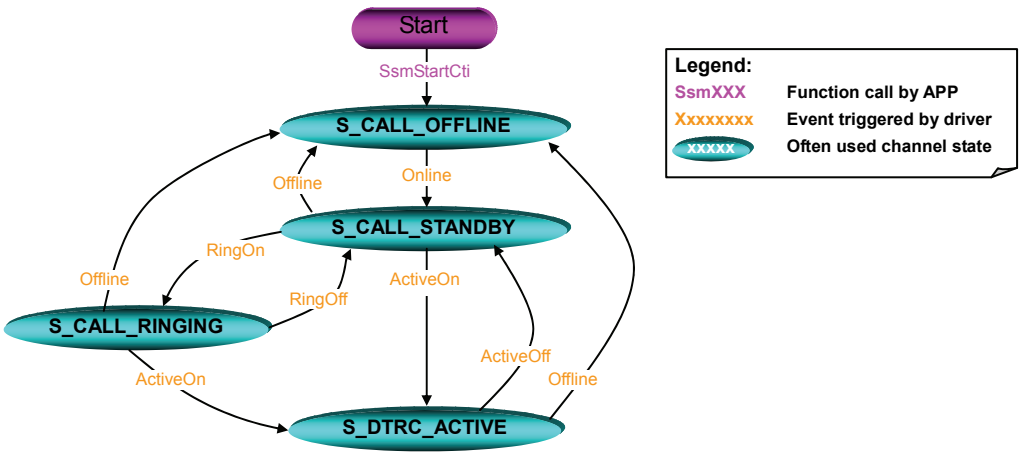
1.19.6.2 使用驱动程序提供的通用呼叫状态机

1.19.6.2.1DST 通道的状态迁移图

驱动程序内部已经实现一个呼叫状态机，它的输入信号是D信道原始信令消息，输出信号为呼叫状态的迁移。状态发生迁移的条件会因为交换机/话机型号的不同而不同。每当DST通道的状态发生迁移时，驱动软件会向应用程序抛出E_CHG_ChState事件。由于驱动程序提供的呼叫状态机反映了呼叫过程的共性特征，因此，它可能无法与特定交换机的呼叫流程完全吻合，但是它却有着非常良好的通用性，几乎适用于任何型号的交换机/话机组合。因此，基于某款型号的交换机/话机开发的应用程序，不必修改代码，只要修改配置项PBXType和PhoneType，就可以用于其它型号的交换机/话机。

虽然通用呼叫状态机可能会丢失一些呼叫细节，例如呼叫保持、呼叫转移、会议、呼入抢接等，但对于大多数的电话录音系统而言，能够获取每次呼叫的起始和结束信息、主叫方/被叫方号码等信息，就已经能够满足要求了。

驱动程序提供的通用呼叫状态机的状态迁移图如下图所示：



DST 状态机的状态转移图

上图中，各状态标识的说明如下表所示：

状态值	宏定义	描述
0	S_CALL_STANDBY	“空闲”状态。
2	S_CALL_RINGING	“振铃”状态。通道状态从 S_CALL_STANDBY 迁移到本状态，表示交换机到话机的来话呼叫。
8	S_CALL_OFFLINE	“断线”状态，板卡与电话机或者交换机的连接被断开。当通道迁移到本状态时，应用程序应该停止录音操作。
128	S_DTRC_ACTIVE	“活动”状态，一般情况下就是通话状态。应用程序可以： ✧ 通道状态从 S_CALL_STANDBY 迁移到 S_DTRC_ACTIVE，表示话机到交换机的去话呼叫； ✧ 通道从其它状态迁移到本状态，可以作为应用程序启动录音操作的触发条件； ✧ 通道从本状态迁移到其它状态，可以作为应用程序停止录音操作的触发条件，并且， ◆ 如果本次呼叫是交换机到话机的来话呼叫，应用程序可以调用函数 SsmGetCallerId或SsmGetCallerIdA获取本次呼叫的主叫方号码； ◆ 如果本次呼叫是话机到交换机的去话呼叫，应用程序可以调用函数 SsmGetPhoNumStr或SsmGetPhoNumStrA获取本次呼叫的被叫方号码。

注：通道的状态值与宏定义均在文件 ShpA3Api.h 中。

驱动程序内部事件的描述如下表所示：

事件名称	描述
Online	板卡与电话机和交换机的连接恢复。
Offline	板卡与电话机或者交换机的连接被断开。
RingOn	驱动程序检测到“开始振铃”事件或者特定的“灯闪烁”事件，因交换机/话机类型而异。
RingOff	驱动程序检测到“停止振铃”事件或者特定的“灯灭”事件，因交换机/话机类型而异。
AcitveOn	驱动程序检测到“开启语音通道”事件、“摘机”事件或者特定的“灯亮”事件，因交换机/话机类型而异。
ActiveOff	驱动程序检测到“关闭语音通道”事件、“挂机”事件或者特定的“灯灭”事件，因交换机/话机类型而异。

1.19.6.2.2如何获取有关电话号码的更多信息

应用程序应在话机到交换机的去话呼叫过程中获取被叫方号码，在交换机到话机的来话呼叫过程中获取主叫方号码。下面分别描述驱动程序如何解析被/主叫方号码，应用程序有哪些获取被/主叫方号码的方法，应用程序开发人员如何验证方法的可行性。

➢ 话机到交换机的去话呼叫

一般情况下，去话呼叫过程中有两次传递被叫方号码的过程：首先，对话机拨号时被叫方号码从话机发送到交换机；然后，交换机发回被叫方号码到话机，话机再将被叫方号码显示在 LCD 屏幕上。下面分别描述。

1) 从话机发送到交换机

话机用户对话机摘机后，可以按数字键（0~9,*,#）拨号，也可以按重拨功能键拨号，具体拨号方法取决于话机用户，下面分别描述。

✧ 话机用户按数字键拨号

话机可能通过 D 信道或 B 信道向交换机发送数字键消息，因交换机/话机类型而异。具体是哪一种情况，可以通过下列方法确定：

✓ 通过阅读交换机的使用说明书确定；

✓ 通过实验方式确定。实验步骤为：

Step 1：搭建测试环境；

Step 2：运行 SynCTI 提供的工具软件 Test.exe（使用方法请参考“SynCTI 应用手册-函数测试工具软件用户手册”）；

Step 3：用话机发起呼叫并按数字键拨号，如果 Test 界面上对应通道的“DKEY”列出现被叫方号码，说明话机使用 D 信道方式；如果 “DTMF 接收缓冲区”列中出现了被叫方号码，说明话机使用 B 信道方式。

下面分别描述话机通过 D/B 信道向交换机发送数字键消息时，获取被叫方号码的方法。

✓ 话机通过 D 信道向交换机发送数字键消息

话机通过D信道向交换机发送数字键按下/弹起消息，驱动程序检测到数字键按下/弹起消息后将消息保存在DKEY缓冲，应用程序可以在检测到通道状态从S_DTRC_ACTIVE迁移到其它状态时调用函数DTRGetDKeyStr获取DKEY缓冲信息作为被叫方号码。

✓ 话机通过 B 信道向交换机发送数字键消息

话机通过B信道向交换机发送DTMF信号，驱动程序检测到DTMF信号后将DTMF保存在DTMF缓冲，应用程序可以在检测到通道状态从S_DTRC_ACTIVE迁移到其它状态时调用函数SsmGetDtmfStr获取DTMF缓冲信息作为被叫方号码。

✧ 话机用户按重拨功能键拨号

话机通过 D 信道向交换机发送重拨功能键消息，驱动程序不能检测到数字键消息，应用程序只能通过其它方法获取被叫方号码。

2) 从交换机发送到话机

交换机收到数字键消息后，通过 D 信道向话机发送 LCD 显示屏更新消息，话机再将号码显示在 LCD 屏幕上。驱动程序检测到 LCD 显示屏更新消息后将消息保存在 LCD 缓冲，在通道状态从 S_DTRC_ACTIVE 迁移到其它状态前一刻将 LCD 缓冲信息拷贝到被叫方号码缓冲。此时，应用程序有两种获取被叫方号码的方法，下面分别描述。

✧ 从 LCD 缓冲获取被叫方号码

在大多数交换机/话机类型下，应用程序可以调用函数DTRGetLCDStr获取LCD缓冲信息，但是应用程序必须选择合适的时机来调用该函数。应用程序开发人员可以通过Test测试后决定获取时机，测试步骤为：

Step 1：搭建测试环境；

Step 2：运行 SynCTI 提供的工具软件 Test.exe（使用方法请参考“SynCTI 应用手册-函数测试工具软件用户手册”）；

Step 3：用话机发起呼叫并按数字键拨号，如果 Test 界面上对应通道的“LCD”列能够显示被叫方号码则此刻可以作为获取时机。

✧ 从被叫方号码缓冲获取被叫方号码

在大多数交换机/话机类型下，应用程序可以在检测到通道状态从S_DTRC_ACTIVE迁移到其它

状态时调用函数SsmGetPhoNumStr或SsmGetPhoNumStrA获取被叫方号码缓冲信息。但是在少数交换机/话机类型下，由于驱动程序将LCD缓冲信息拷贝到被叫方号码缓冲的时机错误，LCD缓冲信息中的被叫方号码已经消失，应用程序用本方法不能获取被叫方号码。应用程序开发人员可以通过Test进行验证，验证步骤为：

Step 1：搭建测试环境；

Step 2：运行 SynCTI 提供的工具软件 Test.exe（使用方法请参考“SynCTI 应用手册-函数测试工具软件用户手册”）；

Step 3：用话机发起呼叫并按数字键拨号，通话后话机挂机，，如果 Test 界面上对应通道的“被叫号码”列能够显示被叫方号码则本方法可用，否则不可用。

调用函数SsmGetPhoNumStr、SsmGetPhoNumStrA或DTRGetLCDStr返回的参数中可能包含时间、日期等信息，应用程序必须使用合适的算法从中分离出被叫方号码。而调用函数DTRGetDKeyStr、SsmGetDtmfStr返回的参数中则没有多余的信息。

➢ 交换机到话机的来话呼叫

一般情况下，交换机对话机响铃的同时在 D 信道上向话机发送 LCD 显示屏更新消息，话机再将号码显示在 LCD 屏幕上。驱动程序检测到 LCD 显示屏更新消息后将消息保存在 LCD 缓冲，并根据交换机/话机类型选择以下时机之一将 LCD 缓冲信息拷贝到主叫方号码缓冲区，

✧ 通道状态从 S_CALL_STANDBY 迁移到 S_CALL_RINGING 前一刻

✧ 通道状态从 S_CALL_RINGING 迁移到 S_DTRC_ACTIVE 前一刻

✧ 通道状态从 S_DTRC_ACTIVE 迁移到其它状态前一刻

此时，应用程序有两种获取主叫方号码的方法，下面分别描述。

✧ 从 LCD 缓冲获取主叫方号码

在大多数交换机/话机类型下，应用程序可以调用函数DTRGetLCDStr获取LCD缓冲信息，但是应用程序必须选择合适的时机来调用该函数。应用程序开发人员可以通过Test测试后决定获取时机，测试步骤为：

Step 1：搭建测试环境；

Step 2：运行 SynCTI 提供的工具软件 Test.exe（使用方法请参考“SynCTI 应用手册-函数测试工具软件用户手册”）；

Step 3：用其它话机呼叫被测话机，通话后话机挂机，，如果 Test 界面上对应通道的“LCD”列能够显示主叫方号码则此刻可以作为获取时机。

✧ 从主叫方号码缓冲获取主叫方号码

在大多数交换机/话机类型下，应用程序可以在检测到通道状态从S_DTRC_ACTIVE迁移到其它状态时调用函数SsmGetCallerId或SsmGetCallerIdA获取主叫方号码缓冲信息。但是在少数交换机/话机类型下，由于驱动程序将LCD缓冲信息拷贝到主叫方号码缓冲的时机错误，LCD缓冲信息中的主叫方号码已经消失，应用程序用本方法不能获取主叫方号码。应用程序开发人员可以通过Test进行验证，验证步骤为：

Step 1：搭建测试环境；

Step 2：运行 SynCTI 提供的工具软件 Test.exe（使用方法请参考“SynCTI 应用手册-函数测试工具软件用户手册”）；

Step 3：用其它话机呼叫被测话机，通话后话机挂机，，如果 Test 界面上对应通道的“CallerId” 列能够显示主叫方号码则本方法可用，否则不可用。

调用函数SsmGetCallerId、SsmGetCallerIdA或DTRGetLCDStr返回的参数中可能包含时间、日期等信息，应用程序必须使用合适的算法从中分离出主叫方号码。

1.19.6.2.3如何获 LCD 上的显示信息

如果需要获取话机LCD显示屏上的显示内容，可以调用函数DTRGetLCDStr，在大多数情况下，该函数返回的信息与电话机的LCD显示屏上显示的内容保持一致。

1.20 ATP 系列板卡（REC 系列）

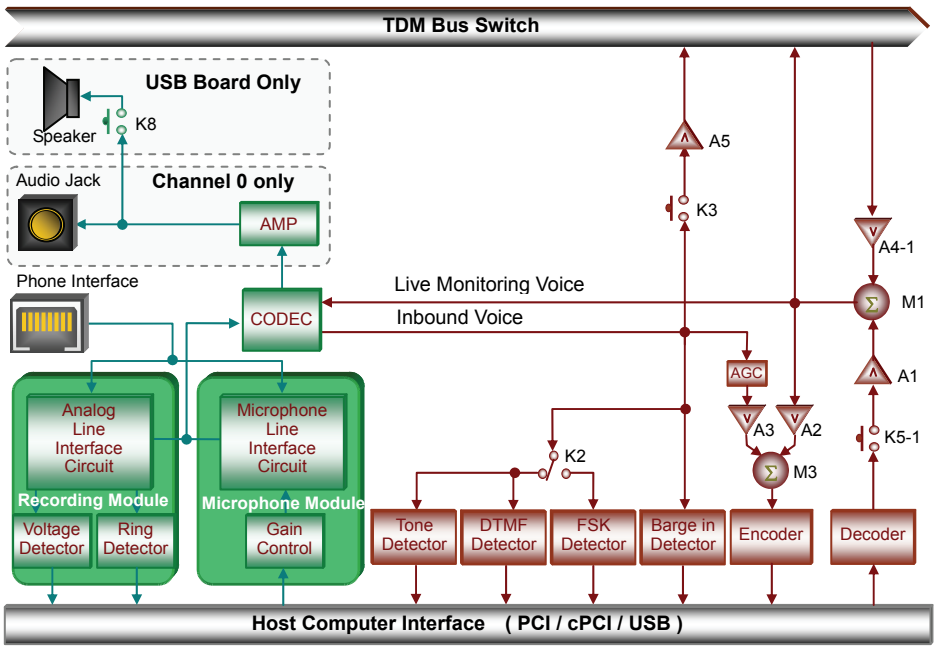
1.20.1 ATP 系列板卡简介

ATP 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Max. Ports	Voltage Detector	CallerID Detector	DTMF Detector	Tone Detector	Energy Detector	TDM Bus	Audio Socket
SHT-2A/USB	USB	2	√	√	√	√	√	N/A	√
SHT-2B/USB	USB	2	√	√	√	√	√	N/A	√
SHT-4A/USB	USB	4	√	√	√	√	√	N/A	√
SHT-4B/USB	USB	4	√	√	√	√	√	N/A	√
SHT-8A/PCI	PCI	8	√	√	√	√	√	N/A	√
SHT-8B/PCI	PCI	8	√	√	√	√	√	N/A	√
SHT-16A-CT/PCI	PCI	16	√	√	√	√	√	H.100	√
SHT-16B-CT/PCI	PCI	16	√	√	√	√	√	H.100	√
SHT-16B-CT/PCI/MP3	PCI	16	√	√	√	√	√	H.100	√
SHT-16B-CT/cPCI	cPCI	16	√	√	√	√	√	H.110	√
SHT-16B-CT/cPCI/MP3	cPCI	16	√	√	√	√	√	H.110	√

1.20.2 ATP 系列板卡的工作原理框图

ATP 系列板卡的工作原理图如下图所示：



原理框图中有关录音操作的符号说明：

名称	描述
M1	实时监听混音器
A4-1	下总线信号的音量调节器
A1	0 通道的放音音量调节器，可通过函数SsmSetPlayVolume/SsmSetPlayGain或配置项DefaultPlayVolume进行设置，缺省值为 0。
K5-1	控制放音操作的开关，由驱动程序自动控制，通常为断开状态，只有当应用程序在 0 通道上调用播放函数时，驱动程序自动开启K5-1 开关；当放音任务结束时，驱动程序自动关闭K5-1。有关语音播放的函数请参见语音播放部分内容。
M3	录音混音器。 M3 的输出进入编码器（Encoder）模块，然后形成录音数据。
A3	来话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecVolume进行设置，也可以通过配置项DefaultRecordVolume设置，缺省值为 0。
A2	去话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecMixer进行设置，也可以通过配置项DefaultRecordMixerVolume设置，缺省值-7，关闭A2。本音量调节器仅用于某些非常特殊的 应用场合。
K3	来话信号是否进入上总线混音器的控制开关，通常为闭合状态，可以通过函数SsmSetFlag（携带F_InVoiceToBus参数）或配置项InVoiceToBus进行设置。
K2	FSK 检测器与其它检测器之间的切换开关，通常指向其它的检测器，只有在开启 FSK 检测器期间才会指向 FSK 检测器。

➢ 通道类型与业务模块

ATP系列板卡可以安装 2 种类型的模块：模拟线路录音模块和麦克风录音模块，但驱动程序并不区分这 2 种模块，它们的通道类型均为模拟录音通道。配置项MicGain用于设置模拟录音通道输入信号的增益。对于模拟中继线录音通道，语音信号来自模拟电话线，应将输入信号的增益设置为正常增益；如果语音信号来自动圈式话筒，应将输入信号的增益设置为 20DB。

➢ **放音操作**

只有 0 通道才支持放音操作。应用程序对 0 通道调用播放函数时，语音数据会被送到板载音频功率放大器。

➢ **录音操作**

ATP 系列板卡的录音操作非常简单，只需要控制 A3 的增益即可。

✧ 范例 1：录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);          // 开启 A3，设置增益为 0DB
SsmRecToFile(ch1, ……);            // 启动文件录音任务……
```

➢ **板载扬声器输出端口**

ATP 系列板卡的 0 通道具有模拟音频放大电路和板载扬声器输出端口，可直接连接到外置的扬声器或耳机，用来对语音信号进行实时监听。其主要用途为：

✧ 将某个通道的来话信号经TDM交换总线直接送到扬声器进行播放。实现此功能的函数为SsmListenTo或SsmListenToEx。

✧ 调用语音播放的函数将应用程序录制的语音文件直接送到板载扬声器进行播放。有关语音播放函数的详细内容请参见本章中语音播放部分。

原理框图中的AMP模块是模拟音频放大电路的功率放大器，它的增益可以通过设函数SsmSetPowerAmpVlm设置。

USB总线的录音盒（型号中带"/USB"标记）的 0 通道除了在板卡上安装了扬声器插座外，还有一个板载扬声器，可以将声音字节播放出来。开关K8 用于选择将 0 通道的输出信号是送到扬声器插座，还是板载扬声器。K8 可以通过函数SsmSetLine0OutTo或配置项USBLine0Output进行设置，缺省为输出到扬声器插座。

注意：对于 8 通道规格的ATP系列板卡，由于板卡上没有TDM交换总线，驱动程序提供了软交换功能，仍然可以将一片板卡上的语音信号送到另外一片板卡（通常为连接了外置扬声器的板卡）上进行实时监听，但需要对配置项BusPlayListen进行设置。

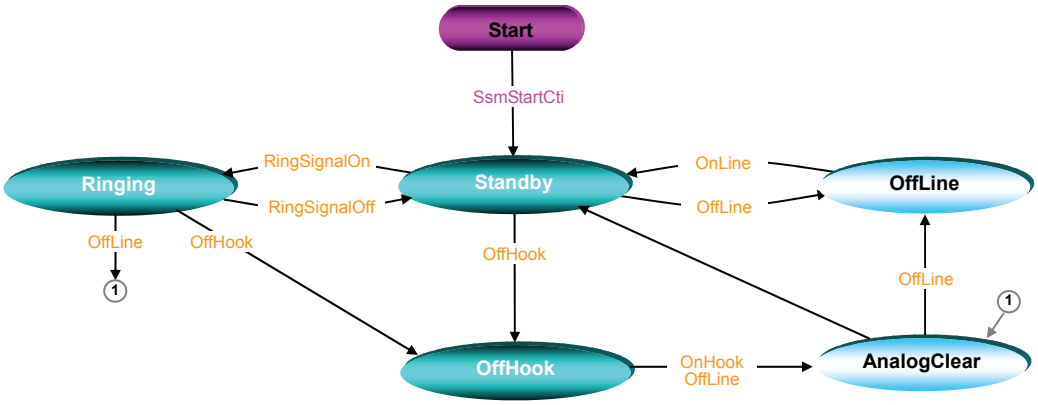
➢ **未安装业务模块的通道的特殊应用**

当ATP系列板卡的通道上没有安装任何业务模块时，可以将此通道作为一个具有录音功能的资源通道使用，此时，通道的Barge in检测器、录音编码器Encoder、下总线操作仍然能够正常工作，从而实现调用录音函数对来自TDM交换总线的语音数据进行录制的功能。Barge in检测器用于提供声控控制。这项特殊应用可以通过配置项NoModuleChBusRec或函数SsmSetNoModuleChBusRec进行设置。

注意：

录音前，应该先调用函数SsmSetRecVolume或设置配置项DefaultRecordVolume将音量调节器A3 关闭。

1.20.3 模拟中继线录音通道的状态迁移图



状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	状态描述
Standby	0	S_CALL_STANDBY	空闲状态。驱动程序初始化成功后，进入本状态。
OffHook	1	S_CALL_PICKUPED	摘机状态。当通道迁移到本状态时，驱动程序会： ✧ 清零并启动信号音检测器； ✧ 启动 DTMF 检测器； ✧ 如果开启了预录音功能，启动预录音操作。
Ringing	2	S_CALL_RINGING	振铃状态，表示有电话正在打入。此时，应用程序可以调用函数SsmGetCallerId或SsmGetCallerIdEx获取主叫号码的信息，更多信息请参见本章中“模拟电话线上的主叫号码”部分内容。
OffLine	8	S_CALL_OFFLINE	线路断开状态。
AnalogClear	10	S_CALL_ANALOG_CLEAR	内部的拆线状态，只持续 8 毫秒，主要用于驱动程序执行一些清理工作，包括： ✧ 清零并关闭信号音检测器； ✧ 如果配置项AutoClearCallerIdBufOnHangup设置为 1，清空驱动程序内部保存Caller ID的缓冲区。 ✧ 如果 Caller ID 的接收制式为 FSK，启动 FSK 制式的 Caller ID 检测器，关闭 DTMF 检测器；否则，启动 DTMF 检测器； ✧ 如果预录音操作仍在进行，立即关闭； ✧ 清空驱动程序内部保存 DTMF 字符的缓冲区； ✧ 启动铃流信号检测器。

驱动程序自动触发的内部事件如下表示：

事件	事件描述
RingSignalOn	驱动程序检测到振铃信号。更多信息请参见本章中“模拟电话线上的铃流信号”部分内容。
RingSignalOff	振铃信号消失。
OffHook	被监测的电话机上发生了摘机操作。更多信息请参见本章中“模拟电话线的电压变化”部分内容。
OnHook	被监测的电话机上发生了挂机操作。
OffLine	被监测的电话机与交换机之间的连接断开，或者高阻复接线到板卡端口的连接断开。更多信息请参见本章中“模拟电话线的电压变化”部分内容。
OnLine	被断开的连接已经恢复。

1.20.4 检测摘机/挂机状态

有关录音通道如何检测摘机动作和挂机动作的详细内容请参见“模拟电话线的电压变化”部分。

1.21 DTP 系列板卡（REC 系列）

1.21.1 DTP 系列板卡简介

DTP 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Trunk Type	Max. Ports	Hi-Z Line Interface Circuit	Voice Processing Capabilities	TDM Bus	DSS1	ISDN PRI	SS7	SS7 Signaling Links
SHD-30A-CT/PCI/FJ	PCI	E1	30	√	√	H.100	√	√	√	1
SHD-60A-CT/PCI/FJ	PCI	E1	60	√	√	H.100	√	√	√	2
SHD-30B-CT/PCI/FJ	PCI	E1/T1	30/23	√	√	H.100	√	√	√	1
SHD-60B-CT/PCI/FJ	PCI	E1/T1	60/46	√	√	H.100	N/A	√	√	2

上表中的语音处理能力（Voice Processing Capabilites）包括：

- ✓ DTMF检测器
- ✓ 信号音检测器
- ✓ Barge in检测器
- ✓ 录音操作
- ✓ 语音播放
- ✓ TDM总线交换

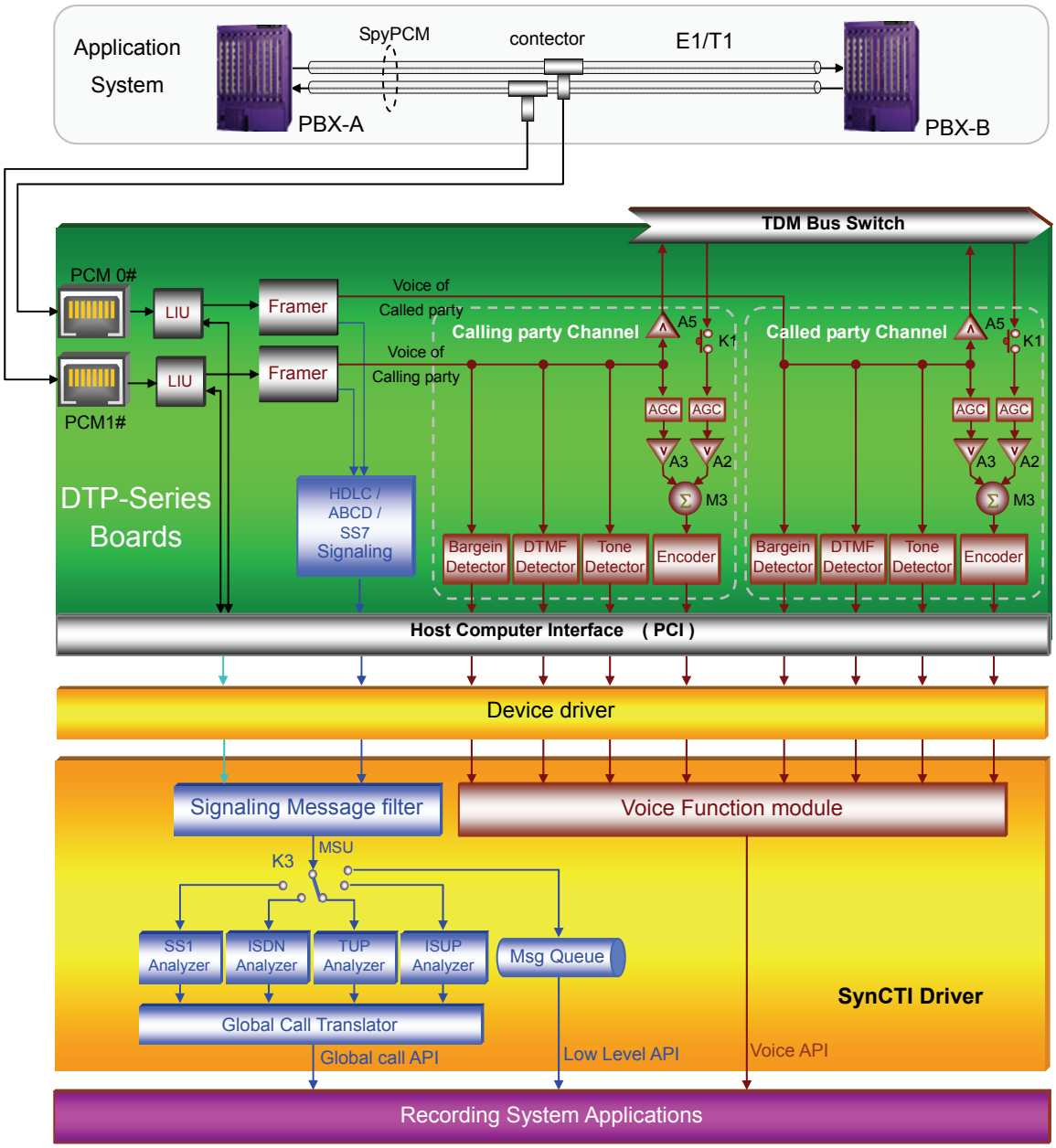
DTP 系列板卡上的线路接口单元（LIU）内置了高阻输入电路，因此可以直接连接被监控的 E1/T1 线路上去，不需要额外的高阻复接设备。

对一个 E1/T1 链路进行监控时，需要同时采集和处理其上行和下行信号，因此需要使用 2 个 E1/T1 接口单元，并且只需要处理输入信号。

获取的监控信息包括 2 部分：信令消息和语音信道数据。

1.21.2 DTP 系列板卡的工作原理框图

DTP 系列板卡的工作原理图如下图所示：



原理框图中与板卡相关的符号说明：

名称	描述
M3	录音混音器。 M3 的输出进入编码器（Encoder）模块，然后形成录音数据。
A3	来话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecVolume进行设置，也可以通过配置项DefaultRecordVolume设置，缺省值为 0。
A2	去话信号进入录音混音器M3 前的音量调节器，可以通过函数SsmSetRecMixer进行设置，也可以通过配置项DefaultRecordMixerVolume设置，缺省值-7，关闭A2。
K3	选择信令或协议的开关，可以通过配置项PcmSSx进行选择。**注意：驱动程序目前还不提供将原始信令消息直接通过Msg Queue传递给应用程序的功能。**

驱动平台已经通过呼叫状态机的方式实现了对 SS1、ISDN、SS7-TUP、SS7-ISUP 信令的分析功能，应用程序可以通过统一的 API 函数（即上图中的 Global Call API）实现对此状态机的控制和访问。本节主要描述与 Global Call API 有关的内容。

对于SS7 信令的ISUP协议和TUP协议，驱动程序实现的呼叫状态机在处理ISUP或TUP消息的同时，如果配置项bOpenSpySS7Msu设置为 1，还可以将收到的原始消息送到一个内部缓冲区中。应用程序可以通过函数SsmGetSs7SpyMsu从该缓冲区中获取原始的ISUP或TUP消息。

语音部分的操作也非常简单。DTP 系列板卡同时为语音信道的来话信号和去话信号配置了独立的语音通道，每个语音信道都具有独立的 DTMF 检测器、信号音检测器、能量检测器、语音编码器、混音录音器等功能。

1.21.3 术语介绍

- 物理 PCM

指板卡上的具有数字中继线处理能力的硬件电路。与 SHD 系列不同的是，DTP 系列的 PCM 处理电路只处理输入信号，却没有输出信号的能力，因此，DTP 系列的卡内 PCM 工作在单工方式。

有关物理PCM的更详细信息请参见本章中"数字中继线"部分。

- 被监控的数字中继线（the PCM line to be monitored， 本书简称 SpyPCM）

SpyPCM 是指被高阻复接的 E1/T1 线路，如上图所示。

由于 E1/T1 线路采用全双工的工作方式（1 收 1 发），因此，为了同时采集和处理 SpyPCM 上的上行和下行信号，在板卡上需要 2 套独立的 PCM 处理电路（即物理 PCM）来分别处理 SpyPCM 的上行和下行信号。这 2 条物理数字中继线及相关的处理电路在逻辑上组合在一起，就能够完全实现对 SpyPCM 上的语音信号和信令消息的采集和处理。

应用程序在调用相关函数时，需要提供SpyPCM的编号，即SpyPCM的逻辑编号。SpyPCM的逻辑编号从 0 开始编号，它与板卡上的 2 条物理PCM之间的绑定关系由配置项TotalSpyPcm和SpyPcm确定。

当与SpyPcm绑定的 2 条物理PCM中的任何 1 条的状态发生变化时，函数SpyGetLinkstatus可以用来获取SpyPcm的状态。

- 监控电路（Circuit in a SpyPCM， 本书简称 SpyCic）

SpyCic 是指 SpyPCM 中的某个具体的时隙（电路）。对于 SS7 信令，SpyCic 就是 TUP 或 ISUP 消息中的 CIC 字段；对于 ISDN PRI 信令和 SS1 信令，SpyCic 就是 PCM 中的时隙编号。

SpyCic 有 2 个编号：物理编号和逻辑编号。SpyCic 的物理编号，是指 SpyCic 在一个 PCM 中的编号，与该 PCM 中的时隙编号有一一对应关系，如下表所示：

SpyCic 的物理编号	0~14	15~29
对应的 PCM 中的时隙编号	1~15	17~31

SpyCic的逻辑编号，是指将整个应用系统中全部PCM的SpyCic从 0 开始进行统一编号。驱动程序提供的关于SpyCic的函数均以SpyCic的逻辑编号作为输入参数。SpyCic的逻辑编号与其物理编号之间的映射关系由配置文件中[AppSpyCICTable]节的配置项TotalAppSpyCIC和AppSpyCIC确定。函数SpyGetMaxCic可以用来获取应用系统中的SpyCic的总数。

由于 1 条 SpyPCM 与 2 条物理 PCM 绑定，因此，SpyCic 的逻辑编号与通道编号存在一一对应关系，从逻辑上看，1 个 SpyCic 由驱动程序中的 2 个通道组成。

- 被叫方通道（Called Party Channel）、主叫方通道（Calling Party Channel）

对于每一次电话呼叫，与 SpyCic 绑定的 2 个通道既可能作为主叫方，也可能作为被叫方。在 SynCTI 驱动程序

实现的信令分析状态机中，处理主叫方送往被叫方的语音信号的通道称为主叫方通道，处理被叫方送往主叫方的语音信号的通道称为被叫方通道。需要注意的是，主叫方通道或被叫方通道并非是固定的，跟呼叫的方向紧密相关。例如，一个通道在某次呼叫过程中是主叫方，因此是主叫方通道；下次呼叫过程中可能是被叫方，因此就会变成被叫方通道。

函数SpyGetCallInCh和SpyGetCallOutCh分别用来获取SpyCic所绑定的 2 个通道在本次呼叫中的被叫方通道编号和主叫方通道编号。函数SpyChToCic可以用来根据通道编号查询其对应的SpyCic的逻辑编号。

SynCTI 驱动程序实现的信令分析状态机均基于 SpyCic，但是应用程序可以对绑定的任何一个通道调用相关的语音处理函数，来获取上行或下行信号中的 DTMF 等信息。

1.21.4 DTP 系列板卡的语音功能

1.21.4.1 获取 DTMF

当基于某个SpyCic的呼叫建立后，调用函数SpyGetCallInCh、SpyGetCallOutCh可以分别获得本次呼叫的被叫方通道号和主叫方通道号，用这2个通道号作为参数调用有关DTMF检测器的相关函数，就可以获得呼叫双方的DTMF信号。有关DTMF检测器的信息请参见本章中“DTMF检测器”部分。

1.21.4.2 语音录制

语音录制功能只支持文件录音方式，相关函数或事件包括：

函数或事件名称	简要描述
SpyRecToFile	基于电路编号的函数，启动文件录音任务
SpyStopRecToFile	基于电路编号的函数，停止文件录音任务
E_PROC_RecordFile	文件录音任务完成后驱动程序抛出的事件

DTP 系列板卡实现录音操作的方式有多种。假设某个 SpyCic 由主叫方通道（记为 CallingPartyCh）和被叫方通道（记为 CalledPartyCh）组成，呼叫双方进入通话状态后，需要实现对此 SpyCic 上的主叫方声音（使用缺省音量）和被叫方声音（使用缺省音量）进行混合录音，基于 SpyCic 的实现代码为：

```
SpyRecToFile(SpyCic, 2, “Voc.wav”, 7, 0, 0xffffffff, 0xffffffff,  0);
```

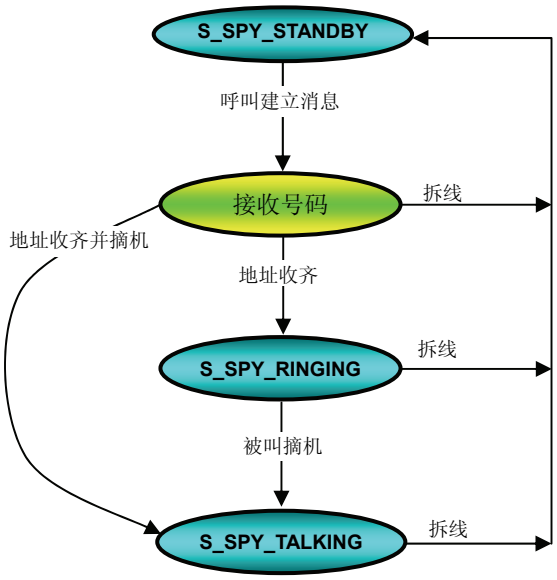
如果需要对录音过程进行更精确的控制，可以使用基于通道的函数来实现。例如，同样实现上述功能，但希望将主叫方声音提升 3DB，将被叫方声音衰减 3DB，则实现代码为：

```
int CalledPartyCh = SpyGetCallInCh(SpyCic);          //获取被叫方通道编号
int CallingPartyCh= SpyGetCallOutCh (SpyCic);        //获取被叫方通道编号
SsmListenTo(CallingPartyCh, CalledPartyCh);          //闭合 CalledPartyCh 上的 K1 开关
SsmSetRecVolume(CalledPartyCh, 1);                    //开启 A3，设置增益为 1*3DB=3DB
SsmSetRecMixer(CalledPartyCh, TRUE, -1);              //开启 A2，设置增益为-1*3DB=-3DB
SsmRecToFile(CalledPartyCh, “Voc.wav”,…);            //启动文件录音任务……
```

1.21.5 驱动程序实现的状态机

1.21.5.1 数字中继线录音通道的通用状态机模型

下图是 DTP 系列板卡使用 SynCTI 驱动程序提供的呼叫状态机的通用状态转移图：



每当SpyCic上的状态发生迁移时，驱动程序都会向应用程序抛出E_CHG_SpyState事件。函数SpyGetState可以用来获取SpyCic的当前状态。

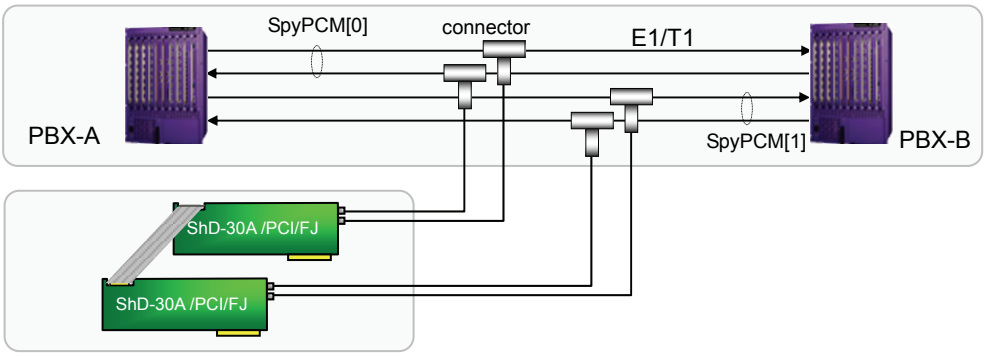
1.21.5.2 获取呼叫信息

有关获取呼叫状态机信息的相关函数或事件包括：

函数或事件名称	简要描述
E_CHG_SpyState	SS1 信令、ISDN 信令、SS7 信令 TUP 协议或 SS7 信令 ISUP 协议的状态机发生状态迁移时，驱动程序抛出的事件
SpyGetState	获取 SpyCic 的状态
SpyGetCallerId	获取本次呼叫的主叫方号码
SpyGetCalleeId	获取本次呼叫的被叫方号码
SpyGetCallInCh	获取本次呼叫的被叫方通道编号
SpyGetCallOutCh	获取本次呼叫的主叫方通道编号

1.21.6 DTP 系列板卡的应用及配置实例

1.21.6.1 使用 No.1 信令/ISDN PRI 协议的系统



上图中，SpyPCM[0]使用 No.1 信令，SpyPCM[1]使用 ISDN 协议，使用 2 片 SHD-30A-CT/PCI/FJ 板卡，则应用程序的配置文件 ShConfig.ini 中的相关配置项可以按照下列方式进行设置：

[BoardId=0]

.....

```
PcmNumber=2

PcmSSx[0]=1                    //No.1 信令
PcmSSx[1]=1                    //No.1 信令
PcmClockMode[0]=2              //从时钟
PcmClockMode[1]=2              //从时钟
```

[BoardId=1]

```
......
PcmNumber=2

PcmSSx[0]=0                    //ISDN 协议
PcmSSx[1]=0                    //ISDN 协议
PcmClockMode[0]=0              //主时钟，线路同步方式
PcmClockMode[1]=2              //从时钟
```

[PcmInfo]

```
TotalPcm=4
Pcm[0]=0,0
Pcm[1]=0,1
Pcm[2]=1,0
Pcm[3]=1,1
```

[SpyPcm]

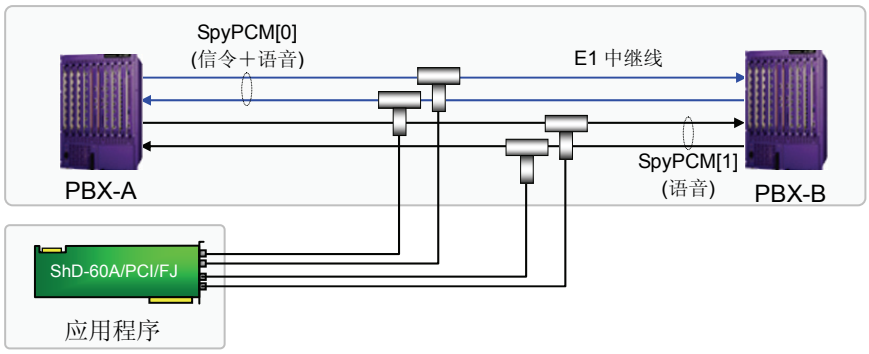
```
TotalSpyPcm=2
SpyPcm[0]=Pcm[0],Pcm[1]
SpyPcm[1]=Pcm[2],Pcm[3]
```

[AppSpyCICTable]

```
TotalAppSpyCIC=60              //总共 60 个 SpyCIC(语音信道)
AppSpyCIC[0]=SpyPcm[0],0..29   //SpyCIC[0]对应于通道 0 和通道 1，依此类推
AppSpyCIC[30]=SpyPcm[1],0..29
```

1.21.6.2 使用 SS7 信令的系统

关于监控 SS7 链路的系统，除配置上述 SS1 或 ISDN 所必需的配置项外，还需增加**[SS7Spy]**一节，具体有以下典型应用：



上图中，PBX-A 与 PBX-B 之间有 2 条数字中继线，SpyPCM[0]的 16 时隙为信令链路，使用 SS7 信令的 TUP 协议，SpyPCM[1]的 16 时隙不使用，其余时隙均作为语音信道使用。以 TUP 协议为例，假设 SpyPCM[0]在 TUP 消息的 CIC 中的数字中继线编号为 4，SpyPCM[1]的编号为 5，则可以按照下述方式设置应用程序的配置文件：

[BoardId=0]

```
......
PcmNumber=4

PcmSSx[0]=7                    //SS7 信令
PcmSSx[1]=7                    //SS7 信令
PcmSSx[2]=7                    //SS7 信令
```

```
PcmSSx[3]=7                 //SS7 信令
PcmClockMode[0]=0           //主时钟，线路同步方式
PcmClockMode[1]=2           //从时钟
PcmClockMode[2]=2           //从时钟
PcmClockMode[3]=2           //从时钟
```

[PcmInfo]

```
TotalPcm=4
Pcm[0]=0,0
Pcm[1]=0,1
Pcm[2]=0,2
Pcm[3]=0,3
```

[SpyPcm]

```
TotalSpyPcm=2
SpyPcm[0]=Pcm[0],Pcm[1]
SpyPcm[1]=Pcm[2],Pcm[3]
```

[AppSpyCICTable]

```
TotalAppSpyCIC=60                   //总共 60 个 SpyCIC(语音信道)
AppSpyCIC[0]=SpyPcm[0],0..29        //SpyCIC[0]对应于通道 0 和通道 1，依此类推
AppSpyCIC[30]=SpyPcm[1],0..29
```

[SS7Spy]

```
TotalSpyLinkSet=1                   //总共只有 1 个信令链路组
SpyLinkSet[0]=SpyPcm[0]             //设置链路组中信令链路的物理位置
SpyLinkSet[0].SpyCICPcm[4]=SpyPcm[0]
SpyLinkSet[0].SpyCICPcm[5]=SpyPcm[1]
```

2 SynCTI 开发包函数说明

2.1 系统函数

2.1.1 驱动平台初始化函数

2.1.1.1 SsmStartCti

SsmStartCti 是整个 SynCTI 驱动程序开发包的入口函数，用来初始化 SynCTI 驱动平台和板卡。

函数原型：

int SsmStartCti(LPSTR *lpSsmCfgFileName*, LPSTR *lpIndexCfgFileName*);

参数说明：

lpSsmCfgFileName	SynCTI 驱动程序的配置文件名（*.INI）。若为空字符串，驱动程序会自动使用其安装目录下的“ShConfig.ini”配置文件。
lpIndexCfgFileName	预加载语音数据的配置文件（*.INI）。若为空字符串，驱动程序会自动使用其安装目录下的“ShIndex.ini”配置文件。

返回值：

-2	初始化失败，因为驱动程序已经装载。
-1	初始化失败，失败原因可以通过SsmGetLastErrMsg函数获得
0	初始化成功

功能描述：

本函数为 SynCTI 驱动程序的入口。本函数将执行下述操作：

- 初始化板卡；
- 如果 lpIndexCfgFileName 不为空，根据配置文件中的信息将预加载语音数据从文件中加载到内存；
- 根据配置文件中语音卡的类型、数量、端口地址、中断号等基本信息，注册和初始化各语音卡，并分配系统资源；
- 为录、放音操作分配系统资源；
- 设置中断服务程序。

注意事项：

- 只有在本函数调用成功后，才能调用本驱动程序中的其它任何函数（SsmGetLastErrMsg 和 SsmGetLastErrCode除外）。
- 应用程序在退出之前，必须调用函数SsmCloseCti关闭本驱动程序以释放驱动程序占用的资源。
- 有关配置文件的详细信息请参见第 4 章。
- 当配置了多块板卡时，如果只有部分板卡初始化成功，驱动程序也会返回 0。应用程序可以通过调用函数 SsmGetMaxCfgBoard和SsmGetMaxUsableBoard来判别是否全部板卡初始化成功。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmCloseCti，SsmGetMaxCfgBoard，SsmGetMaxUsableBoard

2.1.1.2 SsmCloseCti

关闭 SynCTI 驱动程序。

函数原型：

int SsmCloseCti (void)

参数说明：

无

返回值：

1	关闭 SynCTI 驱动程序成功。

功能描述：

关闭 SynCTI 驱动程序。

注意事项：

- 应用程序在退出系统之前，必须调用此函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartCti

2.1.2 设置 API 和事件输出方式的函数

2.1.2.1 SsmSetLogOutput

设置 API 和事件输出的方式。

函数原型：

int SsmSetLogOutput(int nAPIDbg, int nEventStart, int nEventEnd, int nChStart, int nChEnd, DWORD dwReserve)

参数说明：

nAPIDbg	与配置项EnableDebugAPI的取值和功能相同
nEventStart、nEventEnd	与配置项SetEventRange的取值和功能相同
nChStart、nChEnd	与配置项SetChRange的取值和功能相同
dwReserve	保留参数

返回值：

-1	函数不支持。
0	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
1	调用成功。

功能描述：

设置 API 和事件输出的方式。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：

2.1.3 有关定时器的函数

2.1.3.1 SsmStartTimer

向驱动程序申请一个系统定时器。

函数原型：

int SsmStartTimer(WORD wDelay, WORD wMode)

参数说明：

wDelay	定时器的初始值，单位为毫秒，取值范围：0x0008~0xffff
wMode	设置定时器的工作模式： =0(TIMER_ONE)：单次模式，定时器一旦溢出，就自动停止； =1(TIMER_PERIODIC)：周期性模式，每次发生溢出后，由驱动程序自动重新装载初始值wDelay，直到应用程序调用SsmStopTimer函数

返回值：

-1	操作失败
≥0	申请成功，返回定时器的 ID

功能描述：

向驱动程序申请一个系统定时器，并设置该定时器的工作参数。每当定时器发生溢出时，驱动程序会向应用程序抛出E_SYS_TIMEOUT事件，更多内容请参见第 1 章中“使用驱动程序提供的定时器”部分内容。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopTimer，StartTimer

相关事件：E_SYS_TIMEOUT

2.1.3.2 SsmStopTimer

停止一个定时器。

函数原型：

int SsmStopTimer(int nTimer)

参数说明：

int nTimer	定时器的ID，即调用函数SsmStartTimer时的返回值。

返回值：

-1	失败
0	停止成功

功能描述：

停止一个定时器，该定时器由函数SsmStartTimer申请。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartTimer

2.1.3.3 StartTimer

启动通道定时器。

函数原型：

int StartTimer(int ch, WORD wClockID)

参数说明：

ch	通道号
wClockID	定时器 ID，取值范围：0~9

返回值：

-1	失败
0	成功

功能描述：

启动通道定时器。

通道定时器启动后，驱动程序只是将当前时刻保存到变量中，并不会在驱动程序内部判断定时器是否溢出。只有当应用程序调用ElapseTime函数时，驱动程序才会返回定时器的保持时间。

注意事项：

- 本函数启动的定时器不输出任何事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：ElapseTime

2.1.3.4 ElapseTime

获取通道定时器自启动以来的保持时间。

函数原型：

DWORD ElapseTime(int ch, WORD ClockType)

参数说明：

ch	通道号
ClockType:	定时器 ID，取值范围：0~9

返回值：

保持时间，单位为 8 毫秒的倍数。

功能描述：

获取通道定时器自启动以来的保持时间。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：StartTimer

2.1.4 获取系统错误信息的函数

2.1.4.1 SsmGetLastErrCode

获取应用程序调用 API 函数失败时的错误代码。

函数原型：

int SsmGetLastErrCode()

参数说明：

无

返回值：

返回值如下表所示：

返回值	shpa3api.h 中的宏	描述
0	C_ERROR_INIT_FAILURE	初始化失败。
1	C_ERROR_SSMAPI_UNOPENED	驱动程序的 API 接口尚未向应用程序开放。
2	C_ERROR_INVALID_APPCH	非法的通道编号
3	C_ERROR_UNSUPPORTED_OP	不支持的操作。
4	C_ERROR_INDEX_UNOPENED	内存放音功能尚未向应用程序开放。
5	C_ERROR_INVALID_BUSCH	非法的交换总线通道的逻辑编号。
6	C_ERROR_OP_UNOPENED	指定操作没有向应用程序开放。
7	C_ERROR_INVALID_FORMAT	非法语音编码格式。
8	C_ERROR_INVALID_PARAMETER	非法调用参数。
9	C_ERROR_FILEOP_FAILURE	文件操作失败。
10	C_ERROR_MEMORY_FAILURE	内存操作失败。
11	C_ERROR_RESOURCE_USEUP	相关资源耗尽。
12	C_ERROR_SYSTEM	系统错误。
13	C_ERROR_IdleChNotFound	没有空闲通道
14	C_ERROR_OP_FAILURE	操作失败
15	C_ERROR_INVALID_APPSPYCIC	非法的监控电路编号
16	C_ERROR_FAX_NOFILE	传真文件错误
17	C_ERROR_VCH_INVALID_SCALE	参数越界

功能描述：

获取应用程序调用 API 函数失败时的错误代码。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetLastErrMsg

2.1.4.2 SsmGetLastErrMsg

2.1.4.3 SsmGetLastErrMsgA

获取最新一次调用 API 函数失败时的错误信息。

函数原型：

void SsmGetLastErrMsg(LPSTR szErrMsgBuf)

char* SsmGetLastErrMsgA(void)

参数说明：

szErrMsgBuf	存放错误信息字符串的首地址指针，由应用程序分配存储空间，不能少于 300 字节

返回值：

void	无
char*	驱动程序内部保存错误信息字符串的首地址指针

功能描述：

获取最新一次调用 API 函数失败时的错误信息。

注意事项：

- 本函数应该在函数调用失败时立即调用。

相关信息：

驱动程序版本要求	SynCTI Ver.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetLastErrCode

2.1.4.4 fBmp_GetErrMsg

获取最新一次调用 Bmp 函数失败的错误信息。

函数原型：

void fBmp_GetErrMsg(char *buf)

参数说明：

buf	存放错误信息字符串的缓冲区指针

返回值：无。

功能描述：

当调用 Bmp 函数返回失败时，驱动程序将会给出详细的错误信息和错误号，调用该函数可获取该信息。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：

2.1.5 获取板卡信息的函数

2.1.5.1 获取安装板卡的信息

2.1.5.1.1 SsmGetMaxCfgBoard

获取配置文件中设置的 Synway 板卡的总数。

函数原型：

int SsmGetMaxCfgBoard()

参数说明：无

返回值：

-1	调用失败
≥0	返回板卡总数。

功能描述：

获取配置文件中设置的 Synway 板卡的总数。

注意事项：

无。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMaxUsableBoard

2.1.5.1.2 SsmGetMaxUsableBoard

获取驱动程序初始化成功的板卡总数。

函数原型：

int SsmGetMaxUsableBoard()

参数说明：无

返回值：

-1	调用失败
≥0	最大的可用的卡数

功能描述：

获取驱动程序初始化成功的板卡总数。

注意事项：无。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMaxCfgBoard

2.1.5.1.3 GetTotalPciBoard

取得系统中已安装的 PCI 板卡的总数。

函数原型：

int GetTotalPciBoard(int *pTotalBoards)

参数说明：

pTotalBoards	存放总卡数的整型指针，空间由应用程序分配

返回值：

-1	调用失败
0	调用成功

功能描述：

本函数取得系统中已安装的 PCI 板卡的个数。如果调用成功则将板卡个数放到 pTotalBoards 中。

注意事项：

调用本函数无需经过 Shp_a3.dll 来控制，可以直接取得操作系统中已经安装的 PCI 板卡个数。在启动 Shp_a3.dll 之前调用，这样就可以由上层程序修改 ShConfig.ini 配置文件，确保在调用 SsmStartCti 函数时肯定能启动成功。

相关信息：

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数：GetPciBoardSerailNo，GetPciBoardModel

2.1.5.2 获取板卡授权号

2.1.5.2.1 SsmGetAccreditId

2.1.5.2.2 SsmGetAccreditIdEx

获取保存在板卡固件中的客户授权号。SsmGetAccreditId 的返回值中包含了板卡型号和客户授权号信息，SsmGetAccreditIdEx 只返回客户授权号。

函数原型：

int SsmGetAccreditId(int nBId)

int SsmGetAccreditIdEx(int nBId)

参数说明：

nBId	配置文件中指定的板卡 ID 编号。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。	
0	板卡上没有客户授权号的信息。	
>0	SsmGetAccreditId	返回值共有 16 比特，记为 Bit_{15}…Bit_0）： Bit_{15}~Bit_{10}：板卡型号 Bit_9~Bit_0 ：客户授权号

	SsmGetAccreditIdEx	返回客户授权号

功能描述：

获取保存在板卡固件中的客户授权号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetBoardModel

2.1.5.3 获取板卡型号

2.1.5.3.1 SsmGetBoardModel

获取板卡的型号。

函数原型：

int SsmGetBoardModel(int nBId)

参数说明：

nBId	配置文件中指定的板卡 ID 编号。

返回值：

返回值为-1 表示调用失败，失败原因可以通过函数SsmGetLastErrMsg获取；返回值为其它值时，其对应的板卡型号如下表所示：

返回值	对应的板卡型号
0x07	SHD-60A-CT/PCI/SS1
0x09	SHT-16A-CT/PCI
0x0A	SHT-8A/PCI
0x0B	SHD-60A-CT/PCI/SS7
0x0C	SHD-60A-CT/PCI/ISDN
0x0D	SHD-120A-CT/PCI/SS1
0x0E	SHD-120A-CT/PCI/SS7
0x0F	SHD-120A-CT/PCI/ISDN
0x14	SHT-16B-CT/cPCI
0x15	SHD-60A-CT/PCI/FJ
0x16	SHD-30A-CT/cPCI
0x17	SHD-60A-CT/cPCI
0x18	SHD-120A-CT/cPCI
0x1D	SHD-30A-CT/PCI/SS1
0x1E	SHD-30A-CT/PCI/SS7
0x1F	SHD-30A-CT/PCI/ISDN
0x20	SHD-30A-CT/PCI/FJ
0x24	SHT-16B-CT/PCI
0x25	SHT-16B-CT/PCI/FAX 或 SHT-16B-CT/PCI/MP3 [1]
0x26	SHT-8B/PCI
0x27	SHT-8B/PCI/FAX
0x28	SHD-30B-CT/PCI/FAX

0x29	SHD-60B-CT/PCI/FAX
0x2C	SHD-240A-CT/cPCI
0x2D	SHD-480A-CT/cPCI
0x2F	SHD-240S-CT/cPCI
0x30	SHD-480S-CT/cPCI
0x31	SHT-16B-CT/cPCI/MP3
0x32	SHT-16B-CT/cPCI/FAX 或 SHT-16B-CT/cPCI/MP3 [1]
0x36	SHD-60B-CT/cPCI/FAX
0x3b	SHR-16DA-CT/PCI
0x43	DST-1600
0x46	SHT-2A/USB
0x47	SHT-4A/USB
0x4b	SHD-30C-CT/PCI
0x4d	SHD-60C-CT/PCI
0x4f	SHD-30B-CT/PCI/FJ
0x50	SHD-480C-CT/cPCI
0x51	SHD-480C-CT/cPCI/EC
0x52	SHD-240C-CT/cPCI
0x53	SHD-240C-CT/cPCI/EC
0x55	SHV-120A-CT/PCI
0x56	SHV-240A-CT/PCI
0x57	SHD-240D-CT/PCI
0x58	SHD-240D-CT/PCI/EC
0x59	SHD-120D-CT/PCI
0x5a	SHD-120D-CT/PCI/EC
0X5b	SHT-8C/PCI/FAX
0x5c	SHT-16C-CT/PCI/FAX
0x5D	SHN-32A-CT/PCI
0x5f	SHT-2B/USB
0x60	SHT-4B/USB
0x65	SHV-240A-CT/cPCI
0x67	SHD-60B-CT/PCI/FJ
0x6B	SHT-8C/PCI/EC

注[1]：型号为SHT-16B-CT/PCI/FAX和SHT-16B-CT/PCI/MP3 的板卡具有不同的硬件结构，但由于历史的原因，这 2 个型号的板卡上固件中写入的板卡ID号相同，使用配置项DSP3WORKMODE进行区分。

功能描述：

获取板卡的型号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetAccreditIdEx，SsmGetAccreditId

2.1.5.3.2 GetPciBoardModel

获取板卡的型号。

函数原型：

int GetBoardModel(int nBId, LPSTR lpBoardModel)

参数说明：

nBId	配置文件中指定的板卡 ID 编号
lpBoardModel	指定序号的板卡型号，用字符串表示。如下表：

板卡类型＆相对应字符

返回值	含义
“SHD-60A-CT/PCI/SS1”	表示 PCI 60 路 1 号信令卡
“SHD-60A-CT/PCI/SS7”	表示 PCI 60 路 7 号信令卡
“SHD-60A-CT/PCI/ISDN”	表示 PCI 60 路 ISDN 信令卡
“SHD-120A-CT/PCI/SS1”	表示 PCI 120 路 1 号信令卡
“SHD-120A-CT/PCI/SS7”	表示 PCI 120 路 7 号信令卡
“SHD-120A-CT/PCI/ISDN”	表示 PCI 120 路 ISDN 信令卡
“SHD-60B-CT/PCI/FAX”	表示 PCI 60 路 B 型卡，该卡可配置成带传真通道的信令卡
“SHD-30B-CT/PCI/FAX”	表示 PCI 30 路 B 型卡，该卡可配置成带传真通道的信令卡
“SHD-30A-CT/PCI/SS1”	表示 PCI 30 路 1 号 信令卡
“SHD-30A-CT/PCI/SS7”	表示 PCI 30 路 7 号信令卡
“SHD-30A-CT/PCI/ISDN”	表示 PCI 30 路 ISDN 信令卡
“SHD-30A-CT/PCI/FJ”	表示 PCI 30 路 A 型高阻复接卡
“SHD-60A-CT/PCI/FJ”	表示 PCI 60 路 A 型高阻复接卡
“SHD-30B-CT/PCI/FJ”	表示 PCI 30 路 B 型高阻复接卡
“SHD-60B-CT/PCI/FJ”	表示 PCI 60 路 B 型高阻复接卡
“SHT-16B-CT/PCI/FAX 或 SHT-16B-CT/PCI/MP3 [1]”	表示 PCI 16 路 B 型 MP3/FAX 录音卡
“SHT-8B/PCI”	表示 PCI 8 路 B 型模块卡
“SHT-8B/PCI/FAX”	表示 PCI 8 路 B 型带软传真模块卡
“SHT-8C/PCI/FAX”	表示 PCI 8 路 C 型带软传真模块卡
“SHT-8C/PCI/EC”	表示 PCI 8 路 C 型卡
“SHT-16B-CT/PCI”	表示 PCI 16 路 B 型卡
“SHT-16C-CT/PCI/FAX”	表示 PCI 16 路 C 型带软传真模块卡
“SHT-16A-CT/PCI/FAX”	表示 PCI 16 路 A 型带软传真模块卡
“DST-1600”	表示 PCI 16 路 B 型卡
“SHT-16B-CT/cPCI”	表示 cPCI 16 路 B 型模块卡
“SHD-30A-CT/cPCI”	表示 cPCI 30 路卡（支持 SS1、ISDN、SS7）
“SHD-60A-CT/cPCI”	表示 cPCI 60 路卡（支持 SS1、ISDN、SS7）
“SHT-16B-CT/cPCI/MP3 或 SHT-16B-CT/cPCI/FAX”	表示 cPCI 16 路 B 型 MP3/FAX 录音卡
“SHD-480A-CT/cPCI”	表示 cPCI 带会议功能的 16E1 卡
“SHD-480S-CT/cPCI”	表示 cPCI 不带会议功能的 16E1 卡
“SHD-60B-CT/cPCI/FAX”	表示 cPCI 60 路 B 型卡，该卡可配置成带传真通道的信令卡
“SHR-16DA-CT/PCI”	表示 PCI 16 路数字电话录音卡
“SHT-120A-CT/PCI”	表示 PCI 120 路大容量模拟卡
“SHT-2B/USB”	表示 2 路 B 型 USB 模块卡
“SHT-4B/USB”	表示 4 路 B 型 USB 模块卡
“SHT-2A/USB”	表示 2 路 USB 模块卡
“SHT-4A/USB”	表示 4 路 USB 模块卡
“SHT-120A-CT/cPCI”	表示 cPCI 120 路大容量模拟卡
“SHD-240D-CT/PCI”	表示 PCI 带增强回波消除 8 中继数字卡
“SHD-240D-CT/PCI/EC”	表示 PCI 8 中继数字卡
“SHD-120D-CT/PCI”	表示 PCI 带增强回波消除 4 中继数字卡
“SHD-120D-CT/PCI/EC”	表示 PCI 4 中继数字卡
“SHD-480C-CT/cPCI”	表示 cPCI 带增强回波消除 16 中继数字卡
“SHD-480C-CT/cPCI/EC”	表示 cPCI 16 中继数字卡
“SHD-240C-CT/cPCI”	表示 cPCI 带增强回波消除 8 中继数字卡

“SHD-240C-CT/cPCI/EC”	表示 cPCI 8 中继数字卡
“SHD-30C-CT/PCI”	表示 PCI 30 路 C 型卡
“SHD-60C-CT/PCI”	表示 PCI 60 路 C 型卡
“SHN-32A-CT/PCI”	表示 32 路 IP 卡

返回值：

0	调用成功
-1	调用失败

功能描述：

本函数取得指定编号 PCI 板卡的板卡型号。如果调用成功则将板卡型号放到 lpBoardModel 中。

注意事项：

本函数调用无需经过 Shp_a3.dll 来控制，可以直接取得操作系统中已经安装的 PCI 板卡型号。

相关信息：

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数：GetTotalPciBoard，GetPciBoardSerailNo

2.1.5.4 获取板卡序列号

2.1.5.4.1 SsmGetPciSerialNo

获取板卡的序列号。

函数原型：

DWORD SsmGetPciSerialNo(int nBId)

参数说明：

nBId	板卡的 ID 号，对应于配置文件中[BoardId=x]节中的 x 的值

返回值：

0	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
>0	板卡的序列号。

功能描述：

获取板卡的序列号。Synway 板卡在出厂时，会将板卡的序列号贴在板卡的背面上，更多信息请参见板卡的硬件说明手册。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetAccreditIdEx，SsmGetAccreditId

2.1.5.4.2 GetPciBoardSerailNo

获取板卡的序列号。

函数原型：

DWOR GetPciBoardSerailNo (int nBId, DWORD pSerialNo)

参数说明：

nBId	板卡的 ID 号，对应于配置文件中[BoardId=x]节中的 x 的值
pSerialNo	存放 PCI 卡序列号的指针

返回值：

-1	调用失败
0	调用成功号

功能描述：

本函数取得指定 PCI 板卡的序列号。如果调用成功，则将板卡序列号放到 pSerialNo 中。

注意事项：

本函数调用无需经过 Shp_a3.dll 来控制，可以直接取得操作系统中已经安装的 PCI 板卡的序列号。

相关信息：

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数：GetPciBoardModel，GetTotalPciBoard

2.1.5.5 获取文件的版本信息

2.1.5.5.1 SsmGetDllVersion

获取文件“shp_a3.dll”的版本信息。

函数原型：

int SsmGetDllVersion(PSSM_VERSION pDLLVersion)

参数说明：

pDLLVersion	文件版本信息结构指针

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

获取应用程序调用文件“shp_a3.dll”的版本信息。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.1 或更高

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.1.6 获取通道属性的函数

2.1.6.1 SsmGetMaxCh

获取应用系统中的通道总数。

函数原型：

int SsmGetMaxCh(void)

参数说明：无

返回值：

-1	调用失败
≥0	通道总数

功能描述：

获取应用系统中的通道总数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetChType

2.1.6.2 SsmGetChType

获取通道的类型。

函数原型：

int SsmGetChType(int ch)

参数说明：

ch	通道的逻辑编号

返回值：

-1	调用失败
0	模拟中继线通道
2	坐席通道
3	模拟中继线录音通道
4	SS1 通道
5	传真资源通道
6	TUP 通道
7	ISDN 通道（用户侧）
8	ISDN 通道（网络侧）
9	传真资源通道
10	磁石通道
11	ISUP 通道（中国 SS7 信令 ISUP）
12	数字电话线录音通道

15	H.323 通道
16	SIP 通道
20	SHT 系列板卡未安装业务模块的通道

功能描述：

获取通道的类型。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMaxCh

2.1.6.3 SsmGetChHdInfo

获取通道所在板卡的编号，以及通道在板卡内部的物理编号。

函数原型：

int SsmGetChHdInfo(int ch, int * pnBId, int * pnBCh)

参数说明：

ch	通道编号
pnBId	返回 ch 所在的板卡编号，由应用程序分配存储空间。
pnBCh	返回 ch 在板卡上的物理编号，由应用程序分配存储空间。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

获取通道所在板卡的编号，以及通道在板卡内部的物理编号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetAppChId

2.1.6.4 SsmGetAppChId

根据板卡的编号、通道的物理编号查询其逻辑编号。

函数原型：

int SsmGetAppChId(int * ch, int nBId, int nBCh)

参数说明：

ch	返回通道的逻辑编号，由应用程序分配存储空间。
nBId	通道所在板卡的编号
nBCh	通道的物理编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。

0	调用成功

功能描述：

根据板卡的编号、通道的物理编号查询其逻辑编号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetChHdInfo

2.1.7 设置通道属性的函数

2.1.7.1 SsmSetFlag

设置通道的控制属性。

函数原型：

int SsmSetFlag(int ch, int nType, long lPara)

参数说明：

<table>
<tr><td>ch</td><td colspan="3">通道号</td></tr>
<tr><td>nType
lPara</td><td colspan="3">nType 选择需要设置的参数类别，lValue 为参数的值。nType、lValue 的取值范围及物理含义如下表所示：
<table>
<tr><th colspan="2">nType</th><th>描述</th><th>lPara</th></tr>
<tr><th>取值</th><th>shpa3api.h 中的宏</th><th></th><th></th></tr>
<tr><td>1</td><td>F_RCVDTMFSENS</td><td>设置DTMF信号的高电平、低电平的最小保持时间。更多信息请参见配置项ReceiveDtmfSensitive的说明。</td><td>参见配置项ReceiveDtmfSensitive的说明</td></tr>
<tr><td>2</td><td>F_TXDTMFAMP</td><td>设置DTMF发生器产生的DTMF信号的幅度</td><td>高 16 位和低 16 位分别设置组成 DTMF 信号的两个频率的幅度。</td></tr>
<tr><td>3</td><td>F_RCVPHONUMHOLDUP</td><td>设置“被叫号码拦截”功能的控制开关。更多信息请参见配置项PhoNumHoldup的说明。</td><td>取值范围：0/1/2</td></tr>
<tr><td>4</td><td>F_RELATIVEENGYHOOKDETECT</td><td>设置模拟中继线通道上“增强的远端摘机检测器”的工作状态。更多信息请参见第 1 章中“增强的远端摘机检测器”部分内容，或配置项RelativeEngyHookDetect的说明。</td><td>=0：关闭
=1：开启</td></tr>
<tr><td>5</td><td>F_RXR2FILTERTIME</td><td>设置MFC R2 信号的最小保持时间。更多信息请参见配置项RxR2FilterTime的说明。</td><td>取值范围：16~96（必须为 16 的整数倍），单位：毫秒。</td></tr>
<tr><td>6</td><td>F_RECTOFILEA_CALLBACKTIME</td><td>设置函数SsmRecToFileA或SsmRecToFileB的回调时间。更多信息请参见该函数的说明。注意：如果录音格式为GSM、MP3（采用软编码），本参数设置的功能无效</td><td>取值范围：16~4000，单位：毫秒。注意：如果录音格式为 IMA ADPCM，回调时间设置为 128-192（不含 192）ms，将按 128ms 回调；小于 128ms 将按 64ms 回调。</td></tr>
<tr><td>7</td><td>F_CALLERIDSTYLE</td><td>设置对端交换机在模拟电话线上发送主叫号码的模式。更多信息请参见配置项CallerIdStyle的说明。</td><td>=0：DTMF 制式；
=1：FSK 制式。</td></tr>
<tr><td>8</td><td>F_InVoiceToBus</td><td>设置是否将通道的来话信号送到TDM交换总线上，即设置板卡原理框图中k3 开关的状态。更多信息请参见配置项InVoiceToBus的说明。</td><td>=0：关闭
=1：开启</td></tr>
<tr><td>9</td><td>F_ClearInVoiceOnRcvDtmf</td><td>设置DTMF钳位功能。更多信息请参见配置项ClearInVoiceOnRxDtmf的说明。</td><td>=0：关闭
=1：开启</td></tr>
<tr><td>10</td><td>F_MixerResToBus</td><td>设置“下总线信号混音器”的输出信号是否作为“上总线混音器”的输入信号源。本功能只支持 SHT 系列（8/16 通道规格）和 SHD 系列板卡，更多信息请参</td><td>=0：否
=1：是</td></tr>
</table></td></tr>
</table>

			见相关板卡的工作原理框图中的 K1-2 开关。	
	11	F_HighAndLowFreqEnScale	设置DTMF信号中，高频分量部分能量与低频分量部分能量之间的比例。更多信息请参见配置项HighAndLowFreqEnScale的说明。	参见配置项HighAndLowFreqEnScale的说明。
	12	F_DualAndAllFreqEnScale	设置DTMF信号的带内能量占全频能量的百分比判定门限。更多信息请参见配置项DualAndAllFreqEnScale的说明。	参见配置项DualAndAllFreqEnScale的说明。
	13	F_EchoCancelInFsk	当FSK接收器正在工作时，是否关闭回波抵消器。更多信息请参见配置项FskEchoCancelDelay的说明。	=0：关闭； =1：开启。
	14	F_ChToRingingOnRingCnt	设置铃流信号检测器的工作参数。详细信息请参见第 1 章中“模拟电话线上的铃流信号”部分内容，或配置项ChToRingingOnRingCnt的说明。	参见配置项ChToRingingOnRingCnt的说明。
	16	F_SetAdjustCtl	设置回波系数是否进行学习	=0：关闭； =1：开启。

返回值：

-1	调用失败，功能不支持
0	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
1	调用成功

功能描述：

设置通道的控制属性。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetFlag

2.1.7.2 SsmGetFlag

获取通道上的功能开关的状态。

函数原型：

int SsmGetFlag(int ch, int nType, long* plValue)

参数说明：

ch	通道号
nType plValue	nType 选择需要设置的参数类别，具体取值请参见函数的说明。plValue 返回参数的设定值。nType、plValue 的物理含义如下表所示：

nType		描述
取值	shpa3api.h 中的宏	
1	F_RCVDTMFSENS	详细信息请参见函数SsmSetFlag的说明。
2	F_TXDTMFAMP	
3	F_RCVPHONUMHOLDUP	
4	F_RELATIVEENGYHOOKDETECT	
5	F_RXR2FILTERTIME	
6	F_RECTOFILEA_CALLBACKTIME	
7	F_CALLERIDSTYLE	
8	F_InVoiceToBus	
9	F_ClearInVoiceOnRcvDtmf	
10	F_MixerResToBus	
11	F_HighAndLowFreqEnScale	

	12	F_DualAndAllFreqEnScale	
	13	F_EchoCancelInFsk	
	14	F_ChToRingingOnRingCnt	
	15	F_ISDNNet_WaitRemotePickup	plValue 返回驱动程序将通道迁移到 S_CALL_WAIT_REMOTE_PICKUP 状态的具体原因，即收到对端交换机的消息类型： ✧ plValue=4：CALL PROCEEDING 消息； ✧ plValue=5：ALERTING 消息； ✧ plValue=6：CONNECT 消息。 **注意：**本功能要求 SynCTI 的版本为 4.7.1.5 或更高。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

获取通道上的功能开关的状态。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetFlag

2.2 有关事件方式编程的函数

2.2.1 获取事件的函数

2.2.1.1 SsmWaitForEvent

2.2.1.2 SsmWaitForEventA

2.2.1.3 SsmGetEvent

2.2.1.4 SsmGetEventA

获取 Synway 驱动程序输出的事件。SsmGetEvent、SsmGetEventA 是同步函数，不论有没有事件都会立即返回；SsmWaitForEvent、SsmWaitForEventA 是异步函数，如果有事件则立即返回，如果没有事件，调用者线程将被阻塞，直到有事件发生或者超时才返回。

函数原型：

int SsmWaitForEvent(DWORD dwTimeOut, PMESSAGE_INFO pEvent)

int SsmWaitForEventA(DWORD dwTimeOut, PSSM_EVENT pEventEx)

int SsmGetEvent(PMESSAGE_INFO pEvent)

int SsmGetEventA(PSSM_EVENT pEventEx)

参数说明：

dwTimeOut	设置 SsmWaitForEvent 函数的工作模式，取值范围： 0： SsmWaitForEvent 工作在同步模式下，即不管是否有事件，函数都立即返回。此时，SsmWaitForEvent 与 SsmGetEvent 的功能完全一样；

	0xffffffff： 如果没有事件，函数一直挂起，直到有事件时才返回； 其它值： 设置最大等待时间，单位为毫秒。在等待事件期间，如果有事件发生，函数立即返回；如果没有事件发生，等待时间超过本参数的设定值后，函数才返回。
pEvent pEventEx	指向MESSAGE_INFO结构对象(pEvent)或SSM_EVENT结构对象(pEventEx)的指针，用于返回事件信息，仅当函数的返回值为 0 时才有效。更多信息请参见第 1 章中“MESSAGE_INFO”或“SSM_EVENT”部分内容。

返回值：

-1	超时
0	有事件发生，事件信息在 pEvent 或 pEventEx 中返回。

功能描述：

获取 Synway 驱动程序输出的事件。

如果驱动程序内部的事件缓冲区不为空，SsmWaitForEvent 或 SsmWaitForEventA 会立即返回；如果没有事件，本函数会等待事件的发生，直到等待时间超过参数 dwTimeOut 的设定值时，才会返回。

注意事项：

- 只有当驱动程序工作于事件等待模式时才能调用本函数，否则函数将返回失败；
- 如果配置项EventInterfaceType设置为 0，只能调用函数SsmWaitForEvent或SsmGetEvent；如果设置为 1，只能调用函数SsmWaitForEventA或SsmGetEventA，否则函数将返回失败；
- 除了输出事件的数据结构不同以外，函数 SsmWaitForEvent 与 SsmWaitForEventA、SsmGetEvent 与 SsmGetEventA 的功能均完全相同。SsmWaitForEventA 和 SsmGetEventA 通常用于 DST 系列板卡的 D 信道事件编程。

相关信息：

驱动程序版本要求	SsmWaitForEvent、SsmGetEvent 要求 SynCTI Ver. 4.0 或更高； SsmWaitForEventA、SsmGetEventA 要求 SynCTI Ver. 4.7.3.0 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetEvent

2.2.1.5 SsmGetInterEventType

获取配置项EventInterfaceType的值。

函数原型：

int WINAPI SsmGetInterEventType()

参数说明：无

返回值：

-1	失败
0	事件输出参数只能通过MESSAGE_INFO结构获取
1	事件输出参数只能通过SSM_EVENT结构获取

功能描述：

获取配置项EventInterfaceType的值。

注意事项：

- 建议在调用函数SsmSetEvent前先调用本函数，以决定选用哪种结构（MESSAGE_INFO或SSM_EVENT）来获取事件。
- 如果返回值与选用的结构不一致，程序会出现异常。

相关信息：

驱动程序版本要求	SynCTI Ver.4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetEvent

2.2.2 过滤输出事件

2.2.2.1 SsmSetEvent

设置驱动程序的事件输出模式，以及允许或禁止驱动程序输出某个特定事件。

函数原型：

int SsmSetEvent(WORD wEvent, int nReference, BOOL bEnable, PEVENT_SET_INFO pEventSet)

参数说明：

<table>
<tr><td>wEvent</td><td>选择事件，取值范围：
0~0xfffe：事件编码，取值范围请参见第 1 章中“MESSAGE_INFO”部分内容。此时，SsmSetEvent用于通知驱动程序抛出或不抛出特定事件。
0xffff：设置整个驱动程序的工作模式，即事件等待模式和事件回调模式，此时参数 nReference 必须为-1。驱动程序内部的事件队列的深度可以通过配置项 MaxEventPerChannel进行设置。更多信息请参见第 1 章中“设置驱动程序的事件输出模式”部分内容。</td></tr>
<tr><td>nReference</td><td>事件的参考值，wEvent 等于 0xffff 时，本参数必须为-1，用于设置整个驱动程序的工作模式。当参数 wEvent 不等于 0xffff 时，nReference 的物理含义取决于 wEvent 的类型，如下表所示：
<table>
<tr><th>wEvent</th><th>nReference 的含义及取值范围</th></tr>
<tr><td>E_CHG_ISDNStatus
E_CHG_PcmLinkStatus
E_CHG_RemotePCMBlock</td><td>数字中继线的逻辑编号：
nReference=-1：对系统中的全部数字中继线进行设置；
0≤nReference≤M-1：对本参数指定的数字中继线进行设置
注：M 为系统中的数字中继线总数</td></tr>
<tr><td>E_RCV_Ss7Msu</td><td>0-保留</td></tr>
<tr><td>E_CHG_Mtp3State</td><td>DPC 的编号：
nReference=-1：对系统中的全部 DPC 设置；
0≤nReference≤M-1：对本参数指定的 DPC 进行设置</td></tr>
<tr><td>E_SYS_TIMEOUT</td><td>定时器的 ID：
nReference=-1：对系统中的全部定时器设置；
0≤nReference≤M-1：对本参数指定的定时器进行设置</td></tr>
<tr><td>E_CHG_SpyState</td><td>SpyCic 的逻辑编号：
nReference=-1：对系统中的全部被监控电路设置；
0≤nReference≤M-1：对本参数指定的被监控电路进行设置</td></tr>
<tr><td>E_CHG_CICRxPhoNumBuf
E_CHG_CICState
E_PROC_CICAutoDial</td><td>虚电路编号：
nReference=-1：对系统中的虚电路设置；
0≤nReference≤M-1：对本参数指定的虚电路进行设置</td></tr>
<tr><td>其它事件</td><td>通道的逻辑编号：
nReference=-1：对系统中的全部 N 个通道进行设置
0≤nReference≤N-1：对本参数指定的通道进行设置</td></tr>
</table></td></tr>
<tr><td>bEnable</td><td>事件输出的控制标志。当 wEvent=0xffff 且 nReference=-1 时，bEnable=FALSE 有效，用于设置驱动不输出任何事件，bEnable=TRUE 被忽略。当参数 wEvent 不等于 0xffff 时，bEnable 取值范围：
=TRUE：抛出 wEvent；
=FALSE：不抛出 wEvent</td></tr>
<tr><td>pEventSet</td><td>指向具有 EVENT_SET_INFO 结构的对象的指针，用于设置事件的输出条件。如果参数 wEvent=0xffff，本参数不能为 NULL。EVENT_SET_INFO 结构的声明为：
typedef struct _EVENT_SET_INFO
{
DWORD dwWorkMode;</td></tr>
</table>

```
    LPVOID   lpHandlerParam;
    DWORD    dwOutCondition;
    DWORD    dwOutParamVal;
  DWORD    dwUser;
} EVENT_SET_INFO,*PEVENT_SET_INFO;
```

各参数的含义如下表所示：

参数名称：*dwWorkMode*

设置驱动程序抛出事件的模式，只有当参数 wEvent=0xffff 时才有效，wEvent 为其它值时本参数将被忽略。取值范围：

取值	event.h 中的宏	描述
0	NO_EVENT	驱动程序不输出任何事件。
1	EVENT_POLLING	驱动程序工作于事件等待模式，参数 ***dwOutCondition*** 和 ***dwOutParamVal*** 有效，***lpHandlerParam*** 和 ***dwUser*** 无效。
2	EVENT_CALLBACK	驱动程序工作于事件回调模式，此时参数 ***lpHandlerParam*** 和 ***dwUser*** 有效。
3	EVENT_MESSAGE	驱动程序工作于 Windows 消息模式，所有的消息都将被发送到 windows 消息队列中。

参数名称：*lpHandlerParam*

仅当 ***nWorkMode***=EVENT_CALLBACK 或 EVENT_MESSAGE 时有效，***nWorkMode*** 为其它值时，本参数将被忽略。

- ***nWorkMode***=EVENT_CALLBACK

  lpHandlerParam为回调函数指针，回调函数的格式取决于配置项EventInterfaceType的设定值：

  - EventInterfaceType=0：***lpHandlerParam***的格式为：

    int CALLBACK CallBackFun(WORD wEvent,int nReference,DWORD dwParam,DWORD dwUser);

    其中，参数wEvent、nReference和dwParam的含义与MESSAGE_INFO结构中声明的变量的含义完全相同，详细信息请参见函数SsmWaitForEvent或SsmGetEvent的说明。参数dwUser即为调用本函数时应用程序传递给驱动程序的***dwUser***参数。

  - EventInterfaceType=1：***lpHandlerParam***的格式为：

    int CALLBACK CallBackFunA(PSSM_EVENT pEvent);

    其中，pEvent是指向SSM_EVENT结构对象的指针，用于返回事件信息，详细说明请参见函数SsmWaitForEventA或SsmGetEventA。

- ***nWorkMode***=EVENT_MESSAGE

  pHandlerParam 为应用程序接收驱动事件的窗口句柄。

参数名称：*dwUser*

用户自定义参数，仅当 ***nWorkMode***=EVENT_CALLBACK 时有效，***nWorkMode*** 为其它值时，本参数将被忽略。驱动程序会保存此参数，当驱动程序抛出事件时，会将此参数透明地传递给 ***lpHandlerParam*** 指定的回调函数中的同名参数。

参数名称：*dwOutCondition*、*dwOutParamVal*

dwOutCondition设置事件的输出条件，***dwOutParamVal***设置事件输出时携带的参数类别。当事件发生时，驱动程序会将***dwOutParamVal***设定的数据类别放在结构MESSAGE_INFO或SSM_EVENT的dwParam参数中。***dwOutCondition***、***dwOutParamVal***不是必选参数，是否需要设置，取决于参数wEvent的值：

- **wEvent** = E_CHG_LineVoltage：

  当模拟电话线的电压变化幅度超过 **dwOutCondition** 的设定值时，驱动程序才会抛出 E_CHG_LineVoltage 事件。**dwOutParamVal** 保留未用。驱动程序初始化时，会将 **dwOutCondition** 设置为 5。

- **wEvent** = E_PROC_PlayFile：

  dwOutCondition 设置驱动程序抛出 E_PROC_PlayFile 的时间间隔，单位为毫秒。**dwOutParamVal** 设置输出参数的类型，可选值为：

  =0：输出完成播放的百分比；
  =1：输出完成播放的时间；
  =2：输出完成播放的字节总数；
  =3：输出尚未完成播放字节总数。

  驱动程序初始化时，会将 **dwOutCondition** 设置为 1000 毫秒，**dwOutParamVal** 设置为 1。

- **wEvent** = E_PROC_PlayMem：

  - **dwOutParamVal**=END_HALF_BUFFER：
    当驱动程序内部的放音指针越过缓冲区的中间位置或尾部时，抛出 E_PROC_PlayMem 事件。如果放音指针位于缓冲区的前半部分，输出-1；位于后半部分，输出-2；
  - **dwOutParamVal**=END_BUFFER：

	当驱动程序的放音指针越过缓冲区的尾部时，抛出 E_PROC_PlayMem 事件，输出-2。 ◆ **dwOutParamVal**=MEM_OFFSET： 驱动程序累计完成了一定时长的语音数据的播放后，抛出 E_PROC_PlayMem，输出驱动程序内部的放音指针（即在缓冲区中的偏移量）。时间长度由 **dwOutCondition** 指定，单位为毫秒。 ◆ **dwOutParamVal**=MEM_BYTES： 驱动程序累计完成了一定字节数量的语音数据的播放后，抛出 E_PROC_PlayMem，输出驱动程序内部的放音指针（即在缓冲区中的偏移量）。字节数量由 **dwOutCondition** 指定。 驱动程序初始化时，会将 **dwOutParamVal** 设置为 MEM_OFFSET，**dwOutCondition** 设置为 64 字节。 ✧ **wEvent** = E_PROC_RecordFile： **dwOutCondition** 设置驱动程序抛出 E_PROC_RecordFile 事件的时间间隔，单位为毫秒。**dwOutParamVal** 设置输出参数的类型，可选值为： =RECORD_TIME： 输出已经录制完成的时间，单位为毫秒； =RECORD_BYTES：输出已经录制完成的字节总数，单位为字节。 驱动程序初始化时，会将 **dwOutCondition** 设置为 1000 毫秒，**dwOutParamVal** 设置为 RECORD_TIME。 ✧ **wEvent** = E_PROC_RecordMem： ◆ **dwOutParamVal**=END_HALF_BUFFER： 驱动程序内部的录音指针越过录音缓冲区的中间位置或尾部时，驱动程序抛出 E_PROC_RecordMem 事件。如果录音指针位于缓冲区的前半部分，输出-1；位于后半部分，输出-2。 ◆ **dwOutParamVal**=END_BUFFER： 当驱动程序内部的录音指针越过缓冲区的尾部时，抛出 E_PROC_RecordMem 事件，输出-2。 ◆ **dwOutParamVal**=MEM_OFFSET： 驱动程序累计完成了一定时长的语音数据的录音后，抛出 E_PROC_RecordMem，输出驱动程序内部的录音指针（即在缓冲区中的偏移量）。时间长度由 **dwOutCondition** 指定，单位为毫秒。 ◆ **dwOutParamVal**=MEM_BYTES： 驱动程序累计完成了一定字节数量的语音数据的录音后，抛出 E_PROC_RecordMem，输出驱动程序内部的录音指针（即在缓冲区中的偏移量）。字节数量由 **dwOutCondition** 指定。 驱动程序初始化时，会将 **dwOutParamVal** 设置为 MEM_OFFSET，**dwOutCondition** 设置为 64 字节。 ✧ **wEvent** = E_OverallEnergy： 当全频能量的变化幅度超过 **dwOutCondition** 的设定值时，驱动程序才会抛出 E_OverallEnergy 事件。**dwOutParamVal** 保留未用。驱动程序初始化时，会将 **dwOutCondition** 设置为 1000。 ✧ **wEvent** = E_CHG_PeakFrq： 当峰值频率的变化幅度超过 **dwOutCondition** 的设定值时，驱动程序才会抛出 E_CHG_PeakFrq 事件。**dwOutParamVal** 保留未用。驱动程序初始化时，会将 **dwOutCondition** 设置为 100。 ✧ **wEvent** =其它事件： **dwOutParamVal** 和 **dwOutParamVal** 均保留未用，可设置为 0。

返回值：

-1	失败
0	成功

功能描述：

设置驱动程序的事件输出模式，以及允许或禁止驱动程序输出某个特定事件。有关Synway驱动程序的事件输出模式的信息请参见第 1 章中“设置驱动程序的事件输出模式”部分内容。

注意事项：

- SynCTI 驱动程序启动后，会自动工作在轮询模式。如果应用程序使用事件等待方式或事件回调方式进行编程，必须在初始化驱动程序后，调用本函数来改变驱动程序的事件输出模式；
- 如果 wEvent=0xffff，应用程序只能调用一次本函数，并且参数 pEventSet 不能为 NULL，否则函数会返回失败。

相关信息：

驱动程序版本要求	SynCTI Ver.4.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmWaitForEvent， SsmWaitForEventA， SsmGetEvent， SsmGetEventA， SsmPutUserEvent， SsmPutUserEventA

代码范例： 请参见第 1 章中“事件等待方式的编程示例”或“事件回调方式的编程示例”部分内容。

2.2.3 获取编程模式

2.2.3.1 SsmGetEventMode

获取编程模式。

函数原型：

int SsmGetEventMode(WORD wEvent, int nReference, PWORD pwEnable, PEVENT_SET_INFO pEventSet)

参数说明：

wEvent	事件编码： 取值范围请参见第 1 章中“MESSAGE_INFO”部分内容，本参数不能取值为 0xffff。
nReference	事件的参考值： nReference 的物理含义取决于 wEvent 的类型，具体含义同 SsmSetEvent 中参数 nReference，但不能取值为-1。
pwEnable	存放事件输出的控制标志，取值范围： =TRUE： 抛出 wEvent； =FALSE： 不抛出 wEvent
pEventSet	指向具有 EVENT_SET_INFO 结构的对象的指针，用于存放 SsmSetEvent 函数设置的事件输出条件。EVENT_SET_INFO 结构的具体声明见 SsmSetEvent。

返回值：

-1	失败
0	成功

功能描述：

获取应用程序所设置的驱动程序的事件输出模式。

注意事项：

- 当使用本函数获取应用程序的事件编程模式时，wEvent 参数只需传入某个事件，而不支持传入 0xffff。

相关信息：

驱动程序版本要求	SynCTI Ver.4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmSetEvent

2.2.4 通过驱动程序输出应用程序的自定义事件

2.2.4.1 SsmPutUserEvent

2.2.4.2 SsmPutUserEventA

将应用程序的自定义事件添加到驱动程序内部的事件队列中。

函数原型：

int SsmPutUserEvent(WORD wEvent, int nReference, DWORD dwParam)

int SsmPutUserEventA(PSSM_EVENT pEvent)

参数说明：

wEvent	用户自定义的事件编码。**注意：自定义的事件编码不要与驱动程序定义的事件编码重复，建议从 0xfffe 开始取值，逐步递减。**
nReference	事件输出参考值，会通过驱动程序内部的事件队列被透明地回传给应用程序。
dwParam	事件输出参数，会通过驱动程序内部的事件队列被透明地回传给应用程序。
pEvent	指向SSM_EVENT结构对象的指针，用于输入事件信息到驱动程序。pEvent指向的结构对象会通过驱动程序内部的事件队列被透明地回传给应用程序。

返回值：

-1	失败
0	停止成功

功能描述：

将应用程序的自定义事件添加到驱动程序内部的事件队列中。除了使用的事件数据结构不同以外，函数 SsmPutUserEvent 与 SsmPutUserEventA 的功能均完全相同。SsmPutUserEventA 通常用于 DST 系列板卡的 D 信道事件编程。

驱动程序会将应用程序提交的事件先缓存到一个临时缓冲区中，在硬件中断信号到达的时候再将其写入到驱动程序的事件输出队列。配置项MaxUserEventSize用于设置临时缓冲区的深度。

注意事项：

- 本函数只能在“事件等待”的编程模式下调用。有关事件等待模式的信息请参见第 1 章中“设置驱动程序的事件输出模式”部分内容；
- 如果配置项EventInterfaceType设置为 0，只能使用函数SsmPutUserEvent；如果设置为 1，只能使用函数 SsmPutUserEventA，否则函数将返回失败；
- 如果配置项MaxUserEventSize的设定值为 0，本函数会立即返回失败。

相关信息：

驱动程序版本要求	SsmPutUserEvent 要求 SynCTI Ver. 4.0 或更高； SsmPutUserEventA 要求 SynCTI Ver. 4.7.3.0 或更高；
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmWaitForEvent，SsmWaitForEventA，SsmGetEvent，SsmGetEventA

2.3 有关呼叫状态机的函数

2.3.1 有关去话呼叫的函数（CTI 系列）

2.3.1.1 搜索空闲通道

2.3.1.1.1 SsmSearchIdleCallOutCh

在具有去话呼叫能力的通道中搜索一个空闲的通道。

函数原型：

int SsmSearchIdleCallOutCh(WORD wSearchMode, DWORD dwSs7LinkSet)

参数说明：

wSearchMode	指定搜索的模式，可以是下列比特的操作： 比 特 描述

	Bit0	是否在模拟中继线通道中搜索。=1：是；=0：否
	Bit1	是否在 SS1 通道中搜索。=1：是；=0：否
	Bit2	是否在 TUP 通道中搜索。=1：是；=0：否
	Bit3	是否在 ISDN 通道（用户侧）中搜索。=1：是；=0：否
	Bit4	是否在 ISDN 通道（网络侧）中搜索。=1：是；=0：否
	Bit5	是否在 ISUP 通道中搜索。=1：是；=0：否
	Bit6	是否在指定信令链路组的 TUP 通道中搜索。=1：是；=0：否
	Bit7	是否在指定信令链路组的 ISUP 通道中搜索。=1：是；=0：否
	Bit8	是否在 SIP 类型的 IP 通道中搜索。=1：是；=0：否
	Bit9	是否在 H.323 类型的 IP 通道中搜索。=1：是；=0：否
	Bit15~Bit10	保留，应设置为 0
dwSs7LinkSet	TUP 或 ISUP 通道：选择信令链路组； 其它类型通道：　应设置为 0，驱动程序按照从 wSearchMode 的最高位到最低位非零比特的顺序搜索。	

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
≥0	搜索到的通道编号。

功能描述：

在具有去话呼叫能力的通道中搜索一个空闲的通道。驱动程序内部维护了空闲通道的队列。

注意事项：

- 本函数只支持模拟中继线通道、SS1 呼出通道、TUP 通道、ISUP 通道、ISDN 呼出通道、IP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmChkAutoDial

2.3.1.2 发起去话呼叫

2.3.1.2.1 SsmAutoDial

2.3.1.2.2 SsmAutoDialEx

向驱动程序提交 AutoDial 任务，发起一次去话呼叫。SsmAutoDialEx 可以设置 TUP 协议中的主叫用户线标识，通常用于 TUP 通道。

函数原型：

int SsmAutoDial(int ch, LPSTR szPhoNum)

int SsmAutoDialEx(int ch, LPSTR szPhoNum, WORD wParam)

参数说明：

ch	通道编号
szPhoNum	存放被叫方电话号码的缓冲区指针，最大长度为 50。 ● 模拟中继线通道：如果szPhoNum中的字符既不是标准的DTMF字符，也不是指定的闪断字符，则该字符将被视为一个延时字符。有关闪断字符和延时字符的信息请参

<table>
<tr><td></td><td>见函数SsmTxDtmf的说明。
● SS1 通道、TUP/ISUP 通道、ISDN 通道：有效的电话号码必须是‘0’~‘9’之间的数字字符，任何非数字字符将被驱动程序忽略。
● H.323 类型的 IP 通道，格式为：[alias@][transport$]host[:port]
a) alias：缺省与 host 相同
b) transport：缺省为 ip
c) host：可以是主机名、IPv4 格式的地址、GK 注册的别名或 E164 号码
d) port：缺省为 1720
● SIP 类型的 IP 通道，格式为：[“display-name” <][sip:/sips:][user@]host[:port][>]
a) display-name：本端别名，缺省为空
b) sip:/sips:：缺省为 sip:，sips:则是 sip 消息传输加密模式
c) user：用户显示名，缺省为空
d) host：可以是主机名、IPv4 格式的地址、SIP 注册名或 E164 号码
e) port：缺省为 5060</td></tr>
<tr><td>wParam</td><td>本参数只适用于 TUP 通道。
高 4 比特应设置为 0，低 4 比特位（记为 DCBA）有效：
<table>
<tr><th>比特</th><th>含义</th><th>取值</th><th>描述</th></tr>
<tr><td rowspan="4">BA</td><td rowspan="4">主叫号码性质表示语</td><td>00</td><td>市内用户号码</td></tr>
<tr><td>01</td><td>国内备用</td></tr>
<tr><td>10</td><td>国内有效号码</td></tr>
<tr><td>11</td><td>国际号码</td></tr>
<tr><td>C</td><td>提供主叫用户号码表示语</td><td>0</td><td>未限制提供主叫号码</td></tr>
<tr><td></td><td></td><td>1</td><td>限制提供主叫号码</td></tr>
<tr><td rowspan="2">D</td><td rowspan="2">主叫号码不全表示语</td><td>0</td><td>未表示不全</td></tr>
<tr><td>1</td><td>主叫号码不全</td></tr>
</table></td></tr>
</table>

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

向驱动程序提交 AutoDial 任务，发起一次去话呼叫。

AutoDial 任务启动后，通道的状态机开始自动进行去话呼叫接续，相关内容请参见通道的状态转移部分内容。

每当本次去话呼叫有任何进展时，驱动程序会向应用程序抛出E_PROC_AutoDial事件。应用程序也可以在调用SsmChkAutoDial函数对呼叫的进展情况进行查询。由于呼叫进展必然体现到通道的状态迁移上，因此，也可以通过E_CHG_ChState事件或SsmGetChState函数获得通道的状态迁移信息，来获取AutoDial任务的进展情况。

➢ **ISUP 通道**

驱动程序使用 IAM 消息发起去话呼叫。装配一条 IAM 消息的方式有 2 种：

（1） 由驱动程序自动装配。IAM 消息中的部分字段可以通过下列方式进行定制：

设置字段	方法
连接性质表示语[1]	修改配置项DefaultNatureOfConnectionInd的设置。
前向呼叫表示语[1]	修改配置项DefaultIAM_ForwardCallInd的设置。
主叫用户类别[1]	修改配置项DefaultIAM_CAT的设置，或调用SsmSetISUPCAT函数。
传输媒介要求表示语[1]	修改配置项DefaultIAM_TransmissionMediumRequirment的设置。
被叫用户号码[2]	字段中的第 1、第 2 个 8 位位组可以通过配置项DefaultIAM_CalleeParam函数SsmSetIsupFlag（携带ISUP_PhoNumParam参数）设置；后续的 8 位位组（即号码部分）通过本函数的参数szPhoNum设置。
主叫用户号码[3]	字段中的第 1、第 2 个 8 位位组可以通过配置项DefaultIAM_CallerParam或函数SsmSetIsupFlag（携带ISUP_CallerParam参数）设置；后续的 8 位位组（即号码部分）可以通过配置项CalloutCallerId函数SsmSetTxCallerId设置。配置项SetSTSignal用于设置主叫用户号码中是否需要包含ST信号。
原被叫号码[3]	字段中的第 1、第 2 个 8 位位组可以通过配置项DefaultIAM_OriginalCalleeParam设置；后续的 8 位位组（即号码部分）可以通过函数SsmSetTxOriginalCallerID设置。

用户业务信息[3]	配置项bSubscriberSI设置是否包含此字段，配置项SubscriberSI设置此字段的值。
任选前向呼叫表示语[3]	配置项bOptionalFCI设置是否包含此字段，配置项OptionalFCI设置此字段的值。
用户到用户信息[3]	修改配置项Usr2UsrInfo的设置，或者调用函数SsmSetIsupParameter。
其余可选参数[3]	调用函数SsmSetIsupParameter。

注： 表中的 ISUP_CallerParam 和 ISUP_ PhoNumParam 在头文件 Shpa3api.h 中声明，[1]表示长度固定的必备字段，[2]表示长度可变的必备字段，[3]表示可选字段。

（2） 由应用程序自行构建IAM消息。在调用本函数前，应用程序可以自行构建一个IAM消息，然后调用函数SsmSetIsupUPPara将IAM消息提交给驱动程序。驱动程序在执行本函数时，会自动覆盖消息中的SIO、DPC、OPC、SLC、CIC字段，以及被叫用户号码字段。

- **TUP 通道**

TUP 通道可以使用 IAI 消息或 IAM 消息作为初始地址消息发起去话呼叫。有关 IAM/IAI 消息的函数和配置项如下表所示：

消息类型	相关函数或配置项	描述
IAM IAI	配置项：ConnectReqMsg	选择使用 IAM 还是 IAI 消息
	配置项：CalloutIAM_CAT	设置主叫用户类别
	配置项：CalloutIAM_MsgPntr	设置消息表示语字段
IAI	函数：SsmSetTxOriginalCallerID	设置原主叫地址
	配置项：OriginalCalleeAddrInd	设置原被叫地址表示语
	配置项：CalloutCallerId	设置主叫用户号码
	函数：SsmSetTxCallerId	
	配置项：CallingIndicatorBit	设置第 1 表示语八位位组中的附加主叫用户信息表示语和主叫用户线标识表示语
	配置项：SetSTSignal	设置主叫用户号码参数的结尾是否需要包含 ST 信号

如果应用程序在调用本函数前，调用了函数SsmSetTxOriginalCallerID来设置原被叫地址，则驱动程序只使用IAI消息发起去话呼叫。IAI消息中原被叫地址字段中的地址表示语可以通过配置项OriginalCalleeAddrInd进行设置。

- **ISDN 通道**

对于 ISDN 协议，本函数会引发驱动程序使用 SETUP 消息发起去话呼叫。有关 SETUP 消息的函数和配置项如下表所示：

消息类型	相关函数或配置项	描述
SETUP	配置项：UserCalledNoSet，NetCalledNoSet	设置被叫用户号码类型及编号方案
	配置项：UserCallingNoSet，NetCallingNoSet	设置主叫用户号码类型及编号方案
	配置项：UserNumIsFull，NetNumIsFull	设置是否在 SETUP 消息中包含“被叫号码全”参数
	配置项：UserChIdentify，NetChIdentify	设置通路识别表示方法
	配置项：UserChPreference，NetChPreference	设置是否需要优选通路类型
	配置项：CalloutCallerId	设置主叫用户号码（应在调用本函数前调用）
	函数：SsmSetTxCallerId	
	配置项：UserHighLayerCompatible，NetHighLayerCompatible	设置 SETUP 消息中是否包含“高层兼容性”字段
	配置项：UserLowLayerCompatible，NetLowLayerCompatible	设置 SETUP 消息中是否包含“低层兼容性”字段
	函数：SsmISDNSetTxSubAddr	设置主叫用户子地址（应在调用本函数前调用）
	函数：SsmISDNSetDialSubAddr	设置被叫方子地址（应在调用本函数前调用）
	函数：SsmSetTxOriginalCallerID	设置原主叫号码

- **SS1 通道**

对于 SS1 通道，驱动程序在执行 AutoDial 任务时，应用程序可以通过函数或配置项对去话呼叫过程进行局部控制，如下表所示：

相关函数或配置项	描述
函数： SsmSetKA 配置项：MfcKA	设置“主叫用户类别”（即 KA 信号）
函数： SsmSetKD 配置项：MfcKD	设置“发端业务性质”（即 KD 信号）
配置项：CalloutCallerId 函数： SsmSetTxCallerId	设置主叫用户（本端）号码

➢ **IP 通道**

对于 IP 通道，本函数会引发驱动程序使用协议指定的消息发起去话呼叫。具体来说，SIP 类型使用 INVITE 消息，H.323 类型使用 SETUP 消息。有关这些消息的函数和配置项如下表所示：

消息类型	相关函数或配置项	描述
INVITE(SIP)	配置项：GlobalName 函数：SsmSetTxCallerId	设置主叫用户号码（应在调用本函数前调用）
SETUP(H.323)	配置项：GlobalName 函数：SsmSetTxCallerId	设置主叫用户号码（应在调用本函数前调用）

函数执行结果可使用SsmChkAutoDial得到。若遇特殊情况，IP通道拨号动作失败且调用SsmChkAutoDial总是返回 11，用户可调用SsmGetAutoDialFailureReason获得具体失败原因。

注意事项：

- 本函数只支持 TUP 通道、ISUP 通道、SS1 通道、ISDN 通道、模拟中继线通道、IP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAppendPhoNum，SsmChkAutoDial，SsmGetChState，SsmGetPendingReason

2.3.1.2.3 SsmAppendPhoNum

向 AutoDial 任务追加被叫方的电话号码。

函数原型：

int SsmAppendPhoNum(int ch, LPSTR szPhoNum)

参数说明：

ch	通道编号
szPhoNum	指向存放电话号码字符串的缓冲区指针。如果本通道是模拟中继线通道，则如果 szPhoNum 中的字符既不是标准的 DTMF 字符，也不是指定的闪断字符，该字符将被视为一个延时字符；如果是 SS1 的出中继通道，则 szPhoNum 中的任何非数字字符将被驱动程序忽略。拨号字符串的最大长度不得超过 50 个字节。

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

向 AutoDial 任务追加被叫方的电话号码。详细内容请参见相关通道的状态迁移图部分内容。

注意事项：

- 本函数只支持 TUP 通道、ISUP 通道、SS1 通道、ISDN 通道、模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmAutoDialEx

2.3.1.2.4 SsmChkAutoDial

查询 AutoDial 任务的执行进展情况。

函数原型：

int SsmChkAutoDial(int ch)

参数说明：

ch	通道编号

返回值：

-1 表示调用失败，其它返回值的含义如下表所示：

返回值	shpa3api.h 中声明的宏	描述
0	DIAL_STANDBY	通道空闲，没有执行 AutoDial 任务
1	DIAL_DIALING	正在发送被叫号码
2	DIAL_ECHOTONE	回铃。 ✧ 模拟中继线通道：拨号完成后，线路上检测到了回铃音。 ✧ SS1 通道：表明去话呼叫时，驱动程序收到对端交换机的后向 KB=1 或 KB=6 信号，表明被叫空闲。 ✧ TUP/ISUP 通道：表明驱动程序收到对端交换机的地址齐消息(ACM)。
3	DIAL_NO_DIALTONE	没有在线路上检测到拨号音，AutoDial 任务失败。只适用于模拟中继线通道。
4	DIAL_BUSYTONE	被叫用户忙，AutoDial 任务失败。 对于模拟中继线通道，表明线路上检测到了忙音信号；
5	DIAL_ECHO_NOVOICE	拨号完成后，线路上先是出现了回铃音，然后保持静默。AutoDial 任务终止。只适用于模拟中继线通道。
6	DIAL_NOVOICE	拨号完成后，线路上没有检测到回铃音，一直保持静默。AutoDial 任务终止。只适用于模拟中继线通道。
7	DIAL_VOICE	被叫用户摘机，AutoDial 任务完成。
8	DIAL_VOICEF1	被叫用户摘机（检测到 F1 频率的应答信号），AutoDial 任务完成。只适用于模拟中继线通道。
9	DIAL_VOICEF2	被叫用户摘机（检测到 F2 频率的应答信号），AutoDial 任务完成。只适用于模拟中继线通道。
10	DIAL_NOANSWER	被叫用户在指定时间内没有摘机，AutoDial 失败。
11	DIAL_FAILURE	AutoDial 任务失败。失败原因可以通过函数 SsmGetAutoDialFailureReason获得。
12	DIAL_INVALID_PHONUM	被叫用户号码为空号，AutoDial 任务失败。

功能描述：

查询 AutoDial 任务的执行进展情况。

注意事项：

- 坐席通道、磁石通道和录音通道不支持本操作。
- 如果IP通道拨号动作失败，且调用本函数总是获得 11，用户可调用SsmGetAutoDialFailureReason获得具体失败原因。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmAutoDialEx

2.3.1.2.5 SsmGetAutoDialFailureReason

获取去话呼叫失败的具体原因。

函数原型：

int SsmGetAutoDialFailureReason(int ch)

参数说明：

ch	通道编号

返回值：

返回值	宏	描述
-1	-1	操作失败
0	ATDL_NULL	没有呼出任务
1	ATDL_Cancel	AutoDial 任务被应用程序取消
2	ATDL_WaitDialAnsTimeout	等待被叫应答超时
3	ATDL_WaitRemotePickupTimeout	等待被叫摘机超时
10	ATDL_Mtp3Unusable	SS7 信令：信令不可用
11	ATDL_RcvSSB	SS7 信令：收到对端交换机的 SSB 消息
12	ATDL_RcvSLB	SS7 信令：收到对端交换机的 SLB 消息
13	ATDL_RcvSTB	SS7 信令：收到对端交换机的 STB 消息
14	ATDL_RcvUNN	SS7 信令：收到对端交换机的 UNN 消息
15	ATDL_RcvSEC	SS7 信令：收到对端交换机的 SEC 消息
16	ATDL_RcvCGC	SS7 信令：收到对端交换机的 CGC 消息
17	ATDL_RcvNNC	SS7 信令：收到对端交换机的 NNC 消息
18	ATDL_RcvCFL	SS7 信令：收到对端交换机的 CFL 消息
19	ATDL_RcvLOS	SS7 信令：收到对端交换机的 LOS 消息
20	ATDL_RcvSST	SS7 信令：收到对端交换机的 SST 消息
21	ATDL_RcvACB	SS7 信令：收到对端交换机的 ACB 消息
22	ATDL_RcvDPN	SS7 信令：收到对端交换机的 DPN 消息
23	ATDL_RcvEUM	SS7 信令：收到对端交换机的 EUM 消息
24	ATDL_RcvADI	SS7 信令：收到对端交换机的 ADI 消息
25	ATDL_RcvBLO	SS7 信令：收到对端交换机的 BLO 消息
26	ATDL_DoubleOccupy	SS7 信令：检出同抢
27	ATDL_CircuitReset	SS7 信令：收到对端交换机的电路/群复原信号
28	ATDL_BlockedByRemote	SS7 信令：电路被对端交换机闭塞
40	ATDL_SS1WaitOccupyAckTimeout	No.1 信令：等待占用应答信号超时
41	ATDL_SS1RcvCAS_HANGUP	No.1 信令：收到后向拆线信号
42	ATDL_SS1RcvA4	No.1 信令：收到 A4 信号（机键拥塞）
43	ATDL_SS1RcvA5	No.1 信令：收到 A5 信号（空号）
44	ATDL_SS1RcvUndefinedAx	No.1 信令：收到未定义后向 A 信号
45	ATDL_SS1RcvUndefinedAxOnTxCallerId	No.1 信令：送主叫收到未定义 A
46	ATDL_SS1WaitAxTimeout	No.1 信令：等候接收后向 A 组信号超时
47	ATDL_SS1WaitAxStopTimeout	No.1 信令：等候后向 A 组信号停发超时
48	ATDL_SS1WaitAxTimeoutOnTxCallerId	No.1 信令：送主叫时等候 A 信号超时
49	ATDL_SS1WaitAxStopTimeoutOnTxCallerId	No.1 信令：送主叫 A 信号停发超时
50	ATDL_SS1RcvKB2	No.1 信令：收到 KB2 信号(用户“市忙”)
51	ATDL_SS1RcvKB3	No.1 信令：收到 KB3 信号(用户“长忙”)

52	ATDL_SS1RcvKB4	No.1 信令：收到 KB4 信号（机键拥塞）
53	ATDL_SS1RcvKB5	No.1 信令：收到 KB5 信号（空号）
54	ATDL_SS1RcvUndefinedKB	No.1 信令：收到未定义的 KB 信号
55	ATDL_SS1WaitKBTimeout	No.1 信令：接收后向 KB 信号超时
56	ATDL_SS1WaitKBStopTimeout	No.1 信令：等候对方方停发 KB 信号超时
60	ATDL_ISDNNETISBUS	ISDN：网络忙
61	ATDL_ISDNEMPTYNO	ISDN：空号
65	ATDL_IllegalMessage	SS7 信令：收到对端交换机的非法消息
66	ATDL_RcvREL	ISUP：收到对端交换机的释放消息（REL）
67	ATDL_RcvCBK	TUP：收到对端交换机的 CBK 消息
68	ATDL_IPInvalidPhonum	IP：所拨号码无效
69	ATDL_IPRemoteBusy	IP：对端忙
70	ATDL_IPBeenRefused	IP：被拒绝
71	ATDL_IPDnsFail	IP：DNS 无效
72	ATDL_IPCodecUnSupport	IP：不支持的 CODEC 类型
73	ATDL_IPOutOfResources	IP：没有可用的资源
74	ATDL_IPLocalNetworkErr	IP：本端网络出现错误
75	ATDL_IPRemoteNetworkErr	IP：远端网络出现错误
其它		保留

功能描述：

获取去话呼叫失败的具体原因。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmAutoDialEx

2.3.1.2.6 SsmSetTxCallerId

设置去话呼叫时的本端电话号码，即主叫方电话号码。

函数原型：

int SsmSetTxCallerId(int ch, LPSTR pszTxCallerId)

参数说明：

ch	通道编号
pszTxCallerId	● 对于 SIP 类型的 IP 通道，pszTxCallerId 设置的是完整 SIP URI 的 displayname 部份，最大长度不能超过 50。一个完整的 SIP URI 格式为："displayname" <sip:user@host:port>； ● 对于其他类型的通道，该参数存放主叫方电话号码的 ASCII 码形式字符串，最大长度不能超过 50。号码字符必须为数字字符'0'~'9'中的一个，其它类型的字符将被忽略。

返回值：

-1	调用失败
≥0	实际发送的主叫号码个数

功能描述：

设置去话呼叫时本端的电话号码，即主叫方电话号码。

在去话呼叫过程中，如果对端交换机要求本端发送主叫方电话号码，驱动程序会自动将 pszTxCallerId 中的字符发送给对端交换机。

注意事项：

- 本函数必须在函数SsmAutoDial或SsmAutoDialEx之前调用；
- SS1 通道的配置项TxCallerId可以实现相同功能，缺省值为空字符串；
- IP通道的配置项GlobalName可以实现相同功能；
- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、IP 通道；
- 对于TUP协议和ISUP协议，某些交换机要求主叫方发送的主叫号码字符串应包含ST信号（脉冲终了信号）。配置项SetSTSignal可以用来设置此项功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPhoNumLen，SsmAutoDial，SsmAutoDialEx

2.3.1.2.7 SsmGetTxCallerId

获取预设的本端的主叫方电话号码。

函数原型：

int SsmGetTxCallerId(int ch, LPSTR pszTxCallerId)

参数说明：

ch	通道编号
pszTxCallerId	● 对于SIP类型的IP通道，返回通过SsmSetTxCallerId设定的字符串； ● 其他类型的通道，返回主叫号码字符串。

返回值：

-1	调用失败
≥0	pszTxCallerId 中字符的个数

功能描述：

获取预设的本端的主叫方电话号码。

注意事项：

- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、IP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetTxCallerId

2.3.1.2.8 SsmSetTxOriginalCallerID

设置呼叫建立消息中的原主叫号码或原被叫号码。

函数原型：

int SsmSetTxOriginalCallerID(int ch, BYTE* pszTxCallerId)

参数说明：

ch	通道编号
pszTxCallerId	存放原被叫号码（对于 ISUP 协议或 TUP 协议）或原主叫号码（对于 ISDN 协议）的字符串指针，字符总数不能超过 16 个。

返回值：

-1	调用失败
>=0	pszTxCallerId 指示的号码实际长度

功能描述：

去话呼叫时，设置向对端交换机发送的呼叫建立消息中的原主叫号码（对于 ISDN 协议）或原被叫号码（对于 ISUP 协议或 TUP 协议）。

注意事项：

- 本函数必须在函数SsmAutoDial或SsmAutoDialEx之前调用，当驱动程序完成去话呼叫建立消息的装配后，会将此缓冲区清空；
- 对于TUP通道，如果应用程序在调用SsmAutoDial或SsmAutoDialEx前调用了本函数，则驱动程序只使用IAI消息发起去话呼叫。本函数设置IAI消息中原被叫地址字段中原被叫地址，地址表示语可以通过配置项OriginalCalleeAddrInd进行设置；
- 本函数只支持 TUP 通道、ISUP 通道、ISDN 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmAutoDialEx

2.3.1.2.9 SsmSetWaitAutoDialAnswerTime

设置等待被叫用户摘机的最大等待时间。

函数原型：

BOOL SsmSetWaitAutoDialAnswerTime(WORD wSeconds)

参数说明：

wSeconds	等待时间，单位秒

返回值：

FALSE	调用失败
TRUE	调用成功

功能描述：

去话呼叫时，设置等待被叫用户摘机的最大等待时间。

注意事项：

- 配置项MaxWaitAutoDialAnswerTime可以实现相同功能；
- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：SsmGetWaitAutoDialAnswerTime

2.3.1.2.10 SsmGetWaitAutoDialAnswerTime

获取预设的等待被叫用户摘机的最大等待时间。

函数原型：

BOOL SsmGetWaitAutoDialAnswerTime(WORD* pwSeconds)

参数说明：

pwSeconds	返回等待时间，单位秒

返回值：

FALSE	调用失败
TRUE	调用成功

功能描述：

去话呼叫时，获取预设的等待被叫用户摘机的最大等待时间。

注意事项：

- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetWaitAutoDialAnswerTime

2.3.1.2.11 SsmGetKB

获取被叫用户状态。

函数原型：

int SsmGetKB(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	被叫用户状态，具体含义与通道类型及呼叫方向有关： ➢ 去话呼叫： ✧ SS1 通道：返回对端发送本端的KB信号值，具体取值及含义请参见函数SsmSetKB的说明； ✧ TUP 通道：返回从对端交换机收到的 ACM 消息中消息表示语字段，共 8 个比特，具体含义请参见 TUP 协议的相关文档； ✧ ISUP 通道：返回从对端交换机收到的 ACM 消息中后向呼叫表示语字段(共 2 个字节)中的低 8 位比特，具体含义请参见 ISUP 协议的相关文档； ✧ ISDN 通道：不支持。 ➢ 来话呼叫：来话呼叫：返回应用程序调用函数SsmSetKB函数时的KB信号参数值，其取值及含义请参见函数SsmSetKB的说明。

功能描述：

获取被叫用户状态。

注意事项：

- 对于去话呼叫，建议在通道迁移到 S_CALL_WAIT_REMOTE_PICKUP 状态时调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetKB

2.3.1.3 SS1 通道的专用函数

2.3.1.3.1 SsmSetKA

设置去话呼叫时的“主叫用户类别”信号。

函数原型：

int SsmSetKA(int ch, BYTE btSigKA)

参数说明：

ch	通道编号

btSigKA：KA 信号值，取值范围：1≤btSigKA≤10，具体如下表所示：

KA 编码	KA 信号内容（包括 KOA）					
	步进制市话局		纵横制、程控市话局（也包括准电子）			
	KA		KA		KOA	
1	普通	定期	普通	定期	普通	定期
2		用户表，立即		用户表，立即		用户表，立即
3		打印机，立即		打印机，立即		打印机，立即
4	备用		备用		备用	
5	普通免费		普通免费		普通免费	
6	备用		备用		备用	
7	备用		备用		备用	
8	备用		优先，定期		优先，定期	
9	（郊话自动有权、长途自动无权）		备用		备用	
10	（长郊自动无权）		优先，免费		优先，免费	
11			备用			
12						
13			计划用于测试			
14			备用			
15			备用			

返回值：

-1	调用失败
0	调用成功

功能描述：

去话呼叫时，设置R2 信令中的KA信号值（主叫用户类别）。更多信息请参见第 1 章中“中国 1 号信令的状态转移”部分内容。

注意事项：

- 本函数只支持 SHD 系列板卡的 SS1 通道；
- 本函数应该在SsmAutoDial或SsmAutoDialEx函数之前调用；
- 配置项MfcKA可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetKA

2.3.1.3.2 SsmSetKD

去话呼叫时，设置 KD 信号值。

函数原型：

int SsmSetKD(int ch, BYTE btSigKD)

参数说明：

ch	通道编号
btSigKD	KD 信号的编码值，取值范围如下表所示：

KD 值	描述
1	长途话务员半自动呼叫
2	长途自动呼叫（电话通信或用户传真、用户数据通信）
3	市内电话（用户呼叫半自动话务员及国际半自动话务员）
4	市内用户传真或用户数据通信及优先用户
5	半自动核对主叫号码
6	测试呼叫

返回值：

-1	调用失败
0	调用成功

功能描述：

去话呼叫时，设置KD信号值（发端业务性质）。更多信息请参见第 1 章中“中国 1 号信令的状态转移”部分内容。

注意事项：

- 本函数只支持 SHD 系列板卡的 SS1 通道；
- 本函数应该在SsmAutoDial或SsmAutoDialEx函数之前调用；
- 配置项MfcKD可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetKD

2.3.2 有关来话呼叫的函数

2.3.2.1 获取主叫用户信息

2.3.2.1.1 SsmChkOpCallerId

取得指定通道是否支持接收主叫号码信息的操作。

函数原型：

int SsmChkOpCallerId (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	指定通道不支持接收主叫信息操作
1	指定通道支持接收主叫信息操作

功能描述：

取得指定通道是否支持接收主叫号码信息的操作。

注意事项：

- 坐席通道、磁石通道不支持本操作。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetCallerId

2.3.2.1.2 SsmGetCallerId

2.3.2.1.3 SsmGetCallerIdA

2.3.2.1.4 SsmGetCallerIdEx

获取来话呼叫的主叫方号码信息。SsmGetCallerId、SsmGetCallerIdA 用来获取保存在基本缓冲区中的主叫号码信息；SsmGetCallerIdEx 用来获取保存在扩展缓冲区中的主叫号码信息，只适用于模拟中继线通道和模拟中继线录音通道。

函数原型：

int SsmGetCallerId(int ch, LPSTR szCallerId)

int SsmGetCallerIdEx(int ch, LPSTR szCallerIdEx)

char* SsmGetCallerIdA(int ch)

参数说明：

ch	通道编号
szCallerId	存放主叫号码基本信息的字符串指针，由应用程序分配存储空间。
szCallerIdEx	存放主叫号码扩展信息的字符串指针，由应用程序分配存储空间。 **注意：对于模拟中继线通道和模拟中继线录音通道，szCallerIdEx中存放的是FSK制式的主叫号码的原始数据，详细内容请参见第 1 章中“模拟电话线上的主叫号码”部分。**

返回值：

函数名称	返回值	描述
SsmGetCallerId SsmGetCallerIdEx	-1	调用失败
	≥0	主叫号码信息的长度
SsmGetCallerIdA	NULL	调用失败
	其他	驱动程序内部保存主叫号码基本信息的字符串指针

功能描述：

来话呼叫时，获取保存在内部缓冲区中的主叫方号码信息。

对于SS1 通道，如果配置项MfcR2ToRxCallerIdBuf设置为 1，SsmGetCallerIdEx可以取出存放在驱动程序内部的主叫号码扩展信息缓冲区中的对端交换机送出的R2 信号，更多信息请参见第 1 章中“中国 1 号信令的状态机描述”部分内容。

- **IP 通道：**

1) 对 SIP 类型的 IP 通道调用本函数，获得的主叫号码是 SIP URI（SIP 消息中的 From 字段），其格式为：[sip:/sips:]user[@host[:port]]
2) 对 H.323 类型的 IP 通道调用本函数，获得的主叫号码格式为：主机名 (别名) [ip address]。

注意事项：

- 坐席通道、磁石通道不支持本操作。

相关信息：

驱动程序版本要求	SynCTI Ver.4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmClearCallerId，SsmClearCallerIdEx

2.3.2.1.5 SsmGetCallerName

获取模拟电话线上 Fsk 主叫信息中的姓名字符串。

函数原型：

int WINAPI SsmGetCallerName(int ch, LPSTR pszCallerName)

参数说明：

ch	通道编号
pszCallerName	存放姓名字符串的缓冲区指针，由应用程序分配存储空间，不能低于 50 个字符

返回值：

-1	调用失败
≥0	返回 pszCallerName 中字符的个数

功能描述：

获取模拟电话线上Fsk主叫信息中的姓名字符串。更多信息请参见第 1 章中“模拟电话线上的主叫号码”部分内容。

注意事项：

- 本函数只支持 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的模拟电话线录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetCallerId，SsmGetCallerIdA，SsmGetCallerIdEx

2.3.2.1.6 SsmClearCallerId

2.3.2.1.7 SsmClearCallerIdEx

清空驱动程序内部保存主叫号码信息的缓冲区。

函数原型：

int SsmClearCallerId (int ch)

int SsmClearCallerIdEx(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

清空驱动程序内部保存主叫号码信息的缓冲区。SsmClearCallerId 清空基本缓冲区，SsmClearCallerIdEx 清空扩展缓冲区。

注意事项：

- 本函数支持 SHD、ATP、DTP、DST、SHN 系列的全部通道，以及 SHT 系列的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetCallerId，SsmGetCallerIdA，SsmGetCallerIdEx

2.3.2.1.8 SsmGetKA

来话呼叫时，获取主叫用户类别。

函数原型：

int SsmGetKA(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	返回“主叫用户类别”，具体含义与通道类型有关： ✧ SS1 通道：如果本次呼叫为来话呼叫，此返回值即为KA信号值，具体含义请参见函数SsmSetKA的说明；如果为去话呼叫，此返回值表示在本次呼叫过程中，本端发送给对端的“主叫用户类别”； ✧ ISUP 通道：在来话呼叫时，此返回值表示对端交换机的 IAM 消息中的“主叫方类别”参数值，其含义如下表所示：

返回值	注释
9	国内话务员（国内用）
10	普通主叫用户，用于长途局和长途局，以及长途局和市话局之间
11	优先主叫用户
12	数据呼叫（音频带数据）
13	测试呼叫
14	投币电话
15	普通主叫用户，用于市话局之间

✧ TUP 通道：在来话呼叫时，此返回值表示对端交换机的 IAI/IAM 消息中的“主叫方类别”参数值。

功能描述：

来话呼叫时，获取主叫用户类别。

注意事项：

- 本函数只适用于 SS1 通道、TUP 通道和 ISUP 通道；
- 建议应用程序在通道迁移到 S_CALL_RINGING 状态时调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetKA，SsmGetCallerId，SsmGetChState

2.3.2.1.9 SsmGetKD

来话呼叫时，获取发端业务性质。

函数原型：

int SsmGetKD(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	返回主叫方发送的KD信号值，具体取值及含义请参见函数SsmSetKD的说明。

功能描述：

来话呼叫时，获取主叫方的 KD 信号值（即发端业务性质）。

注意事项：

- 本函数只适用于 SHD 系列板卡的 SS1 通道和 DTP 系列板卡的 SpyCic 中被叫方通道；
- 建议在通道迁移到 S_CALL_RINGING 状态时调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmGetChState

2.3.2.2 获取被叫用户信息

2.3.2.2.1 SsmGetPhoNumStr

2.3.2.2.2 SsmGetPhoNumStrA

2.3.2.2.3 SsmGetPhoNumLen

来话呼叫时，获取被叫方号码信息。SsmGetPhoNumLen 获取号码的长度，SsmGetPhoNumStr 和 SsmGetPhoNumStrA 获取具体的号码值。

函数原型：

int SsmGetPhoNumLen(int ch)

int SsmGetPhoNumStr(int ch, LPSTR pszPhoNum)

char* SsmGetPhoNumStrA(int ch)

参数说明：

ch	通道编号
pszPhoNum	存放 ASCII 形式的被叫号码的缓冲区指针，由应用程序分配空间，不小于 50 个字符

返回值：

函数名称	返回值	描述
SsmGetPhoNumStr	-1	调用失败
	≥0	被叫号码的长度
SsmGetPhoNumStrA	NULL	调用失败
	其他	返回驱动程序内部保存被叫号码的缓冲区指针

功能描述：

来话呼叫时，对端交换机会向本端发送呼叫建立消息，驱动程序收到此消息后，会将其中的被叫号码分离出来，保存到内部缓冲区中。本函数用于该缓冲区中的被叫号码信息。

- **IP 通道**

1) 对 SIP 类型的 IP 通道调用本函数，获得的被叫号码是 SIP URI（SIP 消息中的 To 字段），其格式为：["displayname"] [sip:/sips:]user[@host[:port]]；
2) 对 H.323 类型的 IP 通道调用本函数，获得的被叫号码格式为：主机名。

注意事项：

- 本函数只适用于 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、IP 通道上的来话呼叫；
- 当通道迁移到 S_CALL_RING 状态时，调用本函数才能准确地得到本次来话呼叫的被叫号码信息。更多信息请参见第 1 章中相关通道的状态机描述部分内容；

- 驱动程序内部的被叫号码缓冲区长度为 50 个字符，如果收到的被叫号码字符个数超过 50，多余的号码将被丢弃。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：E_CHG_RxPhoNumBuf

2.3.2.3 获取第 1 被叫号码信息

2.3.2.3.1 SsmGet1stPhoNumLen

2.3.2.3.2 SsmGet1stPhoNumStr

2.3.2.3.3 SsmGet1stPhoNumStrA

获取呼叫建立消息中的重定向号码（ISDN 通道）、原被叫号码（TUP 通道）或第一被叫号码/改发号码（ISUP 通道）信息。SsmGet1stPhoNumLen 用于获取信息的长度，SsmGet1stPhoNumStr 和 SsmGet1stPhoNumStrA 用于获取信息本身。

函数原型：

int SsmGet1stPhoNumStr(int ch, LPSTR pszPhoNum)

char *SsmGet1stPhoNumStrA(int ch)

int SsmGet1stPhoNumLen(int ch)

参数说明：

ch	通道编号
pszPhoNum	存放号码字符串的缓冲区指针，由应用程序分配存储空间，不少于 50 个字符

返回值：

函数名称	返回值	描述
SsmGet1stPhoNumStr	-1	调用失败
	≥0	号码字符串的长度
SsmGet1stPhoNumStrA	NULL	调用失败
	其他	返回驱动程序内部保存号码字符串的缓冲区指针

功能描述：

来话呼叫时，获取呼叫建立消息中的被叫号码信息。

- **ISDN 通道**

  获取 SETUP 消息中的重定向号码（Redirection Number）。

- **TUP 通道**

  获取 IAI 或 IAM 消息中的原被叫号码。

- **ISUP 通道**

获取IAM消息中的第一被叫号码信息或改发号码信息。对于ISUP通道，本函数既可以返回IAM消息中的改发号码信息，也可以返回原被叫号码信息，具体是哪一个，取决于配置项SaveRGNTo1stPhoNumStr的设置：

- ✧ SaveRGNTo1stPhoNumStr 设置为 0：如果 IAM 消息中有原被叫号码字段，则返回原被叫号码信息；如果没有，则返回空；
- ✧ SaveRGNTo1stPhoNumStr 设置为 1：如果 IAM 消息中只有改发号码字段，则返回改发号码信息；如果 IAM 消息中只有原被叫号码字段，则返回原被叫号码信息；如果既有改发号码字段，又有原被叫号码字段，则返回排在消息后面的字段中的信息，比如，这 2 个字段在消息中的排列顺序为“改发号码字段|原被叫号码字段”，则返回原被叫号码信息；如果排列顺序为“原被叫号码字段|改发号码字段”，则返回改发号码信息。

如果需要单独获取ISUP通道的IAM消息中的改发号码信息或原被叫号码信息，请使用函数SsmGetIsupParameter。

注意事项：

- 驱动程序内部的缓冲区长度为 50 个字符，当收到的号码数目超过该长度限制时，多余的号码将被丢弃。

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetChState

2.3.2.4 设置来话接续过程参数

2.3.2.4.1 SsmEnableAutoSendKB

设置“自动应答来话呼叫”功能。

函数原型：

int SsmEnableAutoSendKB(int ch, BOOL bEnable)

参数说明：

ch	通道编号
bEnable	=TRUE： 开启 =FALSE：关闭

返回值：

-1	调用失败
0	调用成功

功能描述：

设置“自动应答来话呼叫”功能。

注意事项：

- 本函数只适用于 SHD 系列板卡；
- 对于SS1 入中继通道，本函数设置的参数也可以通过配置项AutoSendKB进行设置，更多内容请参见第 1 章中“中国 1 号信令的状态转移”部分内容；
- 对于ISUP / TUP通道，本函数设置的参数也可以通过配置项AutoSendACM进行设置，更多内容请参见第 1 章中“ISUP通道的状态转移” 或“TUP通道的状态转移”部分内容；
- 对于ISDN通道，本函数设置的参数也可以通过配置项UserSideAutoSendAck（用户侧）或NetSideAutoSendAck（网络侧）进行设置，更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：SsmSetKB，SsmGetAutoSendKBFlag

2.3.2.4.2 SsmGetAutoSendKBFlag

获取“自动应答来话呼叫”功能的标志。

函数原型：

int SsmGetAutoSendKBFlag(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	关闭
1	开启

功能描述：

获取“自动应答来话呼叫”功能的标志。

注意事项：

- 本函数只适用于 SHD 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmEnableAutoSendKB

2.3.2.4.3 SsmSetKB

来话呼叫时，向对端交换机发送后向应答消息，设置被叫用户的状态。

函数原型：

int SsmSetKB(int ch, BYTE btSigKB)

参数说明：

<table>
<tr><td>ch</td><td>通道编号</td></tr>
<tr><td>btSigKB</td><td>被叫状态指示值，具体设置方法与通道类型有关：
➢ SS1 通道：btSigKB 表示向对端交换机发送的后向 KB 信号，其含义为：
<table>
<tr><th rowspan="2">btSigKB</th><th colspan="2">KB 信号内容</th></tr>
<tr><th>收到 KD=1、2 或 6</th><th>收到 KD=3 或 4</th></tr>
<tr><td>1</td><td>被叫用户空闲</td><td>被叫用户空闲
互不控制复原</td></tr>
<tr><td>2</td><td>被叫用户“市忙”</td><td>备用</td></tr>
<tr><td>3</td><td>被叫用户“长忙”</td><td></td></tr>
<tr><td>4</td><td>机键拥塞</td><td>被叫用户忙或机键拥塞</td></tr>
<tr><td>5</td><td>被叫用户为空号</td><td>被叫用户为空号</td></tr>
<tr><td>6</td><td>备用</td><td>被叫用户为空闲
主叫控制复原</td></tr>
</table>
➢ TUP 通道：本参数表示驱动程序发送的 TUP 消息，具体含义为：</td></tr>
</table>

btSigKB	含义	驱动程序发送的消息
1	被叫用户空闲计费	ACM
2	被叫用户“市忙”	SLB
3	被叫用户“长忙”	STB
4	机键拥塞	SEC
5	被叫用户为空号	UNN
6	备用	
7	被叫用户空闲免费	ACM
8	请求主叫号码信息	GRQ（状态会迁移到“等待 GSM”）
9	交换机拥塞信号	SEC
10	电路群拥塞信号	CGC
11	地址不全信号	ADI
12	呼叫故障信号	CFL
13	线路不工作信号	LOS
14	发送专用信息信号	SST
15	接入拒绝信号	ACB
16	不提供数字通路信号	DPN
17	忙	SSB

➢ ISUP 通道：本参数表示驱动程序发送的 ISUP 消息，具体含义为：

btSigKB	含义	驱动程序的操作
1/6	被叫用户空闲，计费	发送 ACM 消息，接受来话呼叫
2	被叫用户忙	发送 REL 消息，拒绝来话呼叫
3	地址不全	发送 REL 消息，拒绝来话呼叫
4	呼叫拒绝	发送 REL 消息，拒绝来话呼叫
5	无应答	发送 REL 消息，拒绝来话呼叫
7	被叫用户空闲，免费	发送 ACM 消息，接受来话呼叫
8	请求主叫号码信息	发送 INR 消息，状态迁移到“等待 INF”
9	用户缺席	发送 REL 消息，拒绝来话呼叫

➢ ISDN 通道：本参数表示驱动程序发送的 ISDN 消息，具体含义为：

btSigKB	含义	驱动程序动作
1	被叫用户空闲	发送 ALERT 消息
2	被叫用户忙	发送 DISCONNECT 消息，原因=被叫用户忙
4	呼叫拒绝	发送 DISCONNECT 消息，原因=呼叫拒绝
5	无应答	发送 DISCONNECT 消息，原因=被叫用户无应答
其他	备用	

返回值：

-1	调用失败
0	调用成功

功能描述：

来话呼叫时，当驱动程序按照预设的收号规则完成被叫号码等信息的接收后，如果应用程序通过函数SsmEnableAutoSendKB或配置项AutoSendACM关闭了“自动应答来话呼叫”功能，通道的状态会迁移到S_CALL_PENDING，将控制权交还给应用程序，由应用程序向对端交换机发送后向应答消息，设置被叫用户的状态。应用程序可以通过调用本函数来接受或拒绝本次来话呼叫。

更多信息请按照通道类型分别参见第 1 章中的相关内容：

✧ SS1 通道：“中国 1 号信令的状态转移”

✧ ISUP通道：“ISUP通道的状态转移”

✧ TUP通道：“TUP通道的状态转移”

✧ ISDN通道：“ISDN通道的状态转移图”

注意事项：

- 本函数只适用于 SHD 系列板卡；
- 对于ISUP通道，如果参数btSigKB设置为 1、6 或 7，驱动程序会向对端交换机应答ACM消息。这个ACM消息的构建方式为：如果应用程序在调用本函数前调用了SsmSetIsupUPPara函数（携带C_ISUP_ACM参数）向驱动程序提交了一个自行构建的ACM消息，使用该ACM消息；否则，使用驱动程序自动构建的ACM消息。更多信息请参见第 1 章中“ISUP通道的状态转移”部分内容。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmEnableAutoSendKB

2.3.3 通道状态迁移事件

2.3.3.1 SsmGetChState

获取通道的当前状态。

函数原型：

int SsmGetChState(int ch)

参数说明：

ch	通道编号

返回值：

返回值-1 表示调用失败，其它返回值的含义如下表所示：

值	shpa3api.h 中的宏	状态描述
0	S_CALL_STANDBY S_FAX_Wait	“空闲”
1	S_CALL_PICKUPED	“摘机”
2	S_CALL_RINGING	“振铃”
3	S_CALL_TALKING	“通话”
4	S_CALL_ANALOG_WAITDIALTONE	模拟中继线通道：去话呼叫，“等待拨号音”
5	S_CALL_ANALOG_TXPHONUM	模拟中继线通道：去话呼叫，“拨号”
6	S_CALL_ANALOG_WAITDIALRESULT	模拟中继线通道：去话呼叫，“等待拨号结果”
7	S_CALL_PENDING	“挂起”状态。函数SsmGetPendingReason可以用来取得挂起原因
8	S_CALL_OFFLINE	“线路断开”状态。
9	S_CALL_WAIT_REMOTE_PICKUP	去话呼叫，“等候被叫用户摘机”。
10	S_CALL_ANALOG_CLEAR	模拟中继线通道：内部状态
11	S_CALL_UNAVAILABLE	“通道不可用”
12	S_CALL_LOCKED	“呼出锁定”
19	S_CALL_RemoteBlock	“对端闭塞”
20	S_CALL_LocalBlock	“本地闭塞”
30	S_CALL_Ss1InWaitPhoNum	SS1 通道：”接收被叫号码”
31	S_CALL_Ss1InWaitFwdStop	SS1 通道：”等待对端交换机停发前向信号”
32	S_CALL_Ss1InWaitCallerID	SS1 通道：”接收 Caller ID”
33	S_CALL_Ss1InWaitKD	SS1 通道：”接收 KD 信号”
34	S_CALL_Ss1InWaitKDStop	SS1 通道：”等待对端交换机停发 KD 信号”
35	S_CALL_SS1_SAYIDLE	SS1 通道：”向对端交换机发送示闲信号”
36	S_CALL_SS1WaitIdleCAS	SS1 通道：”等待对端交换机的示闲信令”
37	S_CALL_SS1PhoNumHoldup	SS1 通道：”号码拦截”状态
38	S_CALL_Ss1InWaitStopSendA3p	SS1 通道：”等待对端交换机停发脉冲方式的 A3 信号”
40	S_CALL_Ss1OutWaitBwdAck	SS1 通道：”等待对端交换机应答占用证实信号”
41	S_CALL_Ss1OutTxPhoNum	SS1 通道：”发送被叫号码”
42	S_CALL_Ss1OutWaitAppendPhoNum	SS1 通道：”等待应用程序追加电话号码”
43	S_CALL_Ss1OutTxCallerID	SS1 通道：”发送主叫号码”
44	S_CALL_Ss1OutWaitKB	SS1 通道：”等待对端交换机的 KB 信号”
45	S_CALL_Ss1OutDetectA3p	SS1 通道：”等待对端交换机的 A3 脉冲信号”
50	S_FAX_ROUND	FAX 通道：”状态转移过程中”
51	S_FAX_PhaseA	FAX 通道：”传真呼叫建立”（PhaseA）
52	S_FAX_PhaseB	FAX 通道：”传真报文前处理”（PhaseB）

53	S_FAX_SendDCS	FAX 通道：”传真发送中向接收方发送 DCS 信号”
54	S_FAX_Train	FAX 通道：”传真报文传输前传输训练”
55	S_FAX_PhaseC	FAX 通道：”传真报文传输中”（PhaseC）
56	S_FAX_PhaseD	FAX 通道：”传真报文后处理”(PhaseD)
57	S_FAX_NextPage	FAX 通道：”传真报文传输下一页”
58	S_FAX_AllSent	FAX 通道：”传真发送中报文传输结束”
59	S_FAX_PhaseE	FAX 通道：”传真呼叫释放”(PhaseE)
60	S_FAX_Reset	FAX 通道：”复位 MODEM”
61	S_FAX_Init	FAX 通道：”初始化 MODEM”
62	S_FAX_RcvDCS	FAX 通道：传真接收，”接收发方的 DCS 信号”
63	S_FAX_SendFTT	FAX 通道：传真接收，”发送训练失败信号 FTT”
64	S_FAX_SendCFR	FAX 通道：传真接收，”发送可接受的证实信号 CFR”
70	S_TUP_WaitPcmReset	TUP 通道：“电路复原”
71	S_TUP_WaitSAM	TUP 通道：“等待对端交换机的后续地址消息”
72	S_TUP_WaitGSM	TUP 通道：“等待对端交换机的 GSM 消息”
73	S_TUP_WaitCLF	TUP 通道：“等待对端交换机的拆线消息”
74	S_TUP_WaitPrefix	TUP 通道：“接收入局字冠”
75	S_TUP_WaitDialAnswer	TUP 通道：“等待对端交换机的应答消息”
76	S_TUP_WaitRLG	TUP 通道：“等待对端交换机的 RLG 消息”
77	S_TUP_WaitSetCallerID	TUP 通道："等待应用程序设置主叫号码"
81	S_ISDN_OUT_WAIT_NET_RESPONSE	ISDN 通道：“等待网络响应”
82	S_ISDN_OUT_PLS_APPEND_NO	ISDN 通道：“等待应用程序追加号码”
83	S_ISDN_IN_CHK_CALL_IN	ISDN 通道：“检测到呼入”
84	S_ISDN_IN_RCVING_NO	ISDN 通道：“正在接收号码”
85	S_ISDN_IN_WAIT_TALK	ISDN 通道：“准备进入通话”
86	S_ISDN_OUT_WAIT_ALERT	ISDN 通道：“等待对方发提醒信号”
87	S_ISDN_CALL_BEGIN	ISDN 通道：“发起呼叫”(去话)，或”检测到呼入”(来话)
88	S_ISDN_WAIT_HUANGUP	ISDN 通道：“等待释放完成”
100	S_CALL_SENDRING	磁石通道：“发送振铃”
120	S_ISUP_WaitSAM	ISUP 通道：“等待对端交换机的 SAM”
121	S_ISUP_WaitRLC	ISUP 通道：“等待对端交换机的释放监护信号 RLC”
122	S_ISUP_WaitReset	ISUP 通道：“电路复原”
123	S_ISUP_LocallyBlocked	ISUP 通道：“本地闭塞”
124	S_ISUP_RemotelyBlocked	ISUP 通道：“远端闭塞”
125	S_ISUP_WaitDialAnswer	ISUP 通道：“等待对端交换机的应答消息”
126	S_ISUP_WaitINF	ISUP 通道：“等待对端交换机的 INF 消息”
127	S_ISUP_WaitSetCallerID	ISUP 通道：“等待应用程序设置主叫号码”
128	S_DTRC_ACTIVE	DTR 通道：“被监控话路处于非空闲状态“
129	S_ISUP_Suspend	ISUP 通道：“暂停“
130	S_CALL_EM_TXPHONUM	E/M 通道：”拨号”
131	S_CALL_EM_WaitIdleCAS	E/M 通道：”等待对端的示闲信号”
132	S_CALL_VOIP_DIALING	IP 通道：VoIP 主叫拨号状态
133	S_CALL_VOIP_WAIT_CONNECTED	IP 通道：VoIP 被叫摘机等待进入通话状态
134	S_CALL_VOIP_CHANNEL_UNUSABLE	IP 通道：VoIP 通道目前不可用
其它		保留

功能描述：

获取通道的当前状态。

注意事项：

- 为提高应用程序的运行效率，建议使用 E_CHG_ChState 事件来获取通道状态的变化信息。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetChStateKeepTime，SsmGetPendingReason

2.3.3.2 SsmGetChStateKeepTime

获取通道在当前状态的保持时间。

函数原型：

long SsmGetChStateKeepTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	本通道在当前状态的停留时间，单位为毫秒

功能描述：

获取通道在当前状态的保持时间。

灵活使用本函数可以得到通道的许多信息，例如：

- 当呼入通道处于 S_CALL_RINGING 状态时，本函数可以取得振铃持续的时间；
- 当本通道处于 S_CALL_TALKING 状态时，本函数可以取得通话持续的时间；
- 当呼出通道处于 S_CALL_WAIT_REMOTE_PICKUP 状态时，本函数可以取得等待被叫用户摘机时的等待时间。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmGetChState

2.3.4 获取 PendingReason

2.3.4.1 SsmGetPendingReason

获取通道进入 S_CALL_PENDING 状态的具体原因。

函数原型：

int SsmGetPendingReason(int ch)

参数说明：

ch	通道编号

返回值：

-1 表示调用失败，其它值的含义如下表所示：

返回值	宏	适用通道类型	描述
0	ANALOGOUT_NO_DIALTONE	模拟中继线通道	去话呼叫失败：未检测到拨号音
1	ANALOGOUT_BUSYTONE	模拟中继线通道	去话呼叫失败：检测到忙音
2	ANALOGOUT_ECHO_NOVOICE	模拟中继线通道	去话呼叫：检测到回铃音后，线路上保持静默，驱动程序无法判别被叫是否摘机
3	ANALOGOUT_NOANSWER	模拟中继线通道	去话呼叫：检测到回铃信后，被叫用户未在配置项 MaxWaitAutoDialAnswerTime 指定的时间内应答。
4	ANALOGOUT_TALKING_REMOTE_HANGUPED	模拟中继线通道	通道在“连接”状态时检测到对端用户挂机
5	ANALOGOUT_NOVOICE	模拟中继线通道	去话呼叫：拨号过程已经完成，未在线路上检测到回铃音

			和任何其它的语音信号，驱动程序无法判断被叫用户是否摘机
10	PEND_WaitBckStpMsg	ISUP / TUP ISDN SS1	来话呼叫：驱动程序按照预设的收号规则完成被叫号码等信息的接收后，“自动应答来话呼叫”功能未开启，需要应用程序自行决定是否接受此呼叫。
11	SS1IN_BWD_KB5	SS1	来话呼叫：收到对端交换机的 KD 信号后，应用程序设置 KB=5（空号），等待主叫方的拆线信号
12	PEND_RemoteHangupOnTalking	ISUP / TUP SS1	来话呼叫：双方通话时，对端挂机
13	PEND_AutoDialFailed	ISUP / TUP	去话呼叫失败。具体的自动拨号失败原因，可调用 SsmGetAutoDialFailureReason()获得。
14	PEND_SsxUnusable	ISUP / TUP	SS7 信令不可用
15	PEND_CircuitReset	ISUP / TUP	发生电路复原事件
16	PEND_PcmSyncLos	ISUP / TUP SS1	数字中继线的基本帧（0 时隙）同步信号丢失
20	SS1OUT_TALKING_REMOTE_HANGUPED	SS1	去话呼叫：双方通话时，对端挂机
	PEND_CalleeHangupOnTalking	ISUP / TUP	
21	SS1OUT_NOANSWER	SS1	去话呼叫失败：被叫用户未在配置项 MaxWaitAutoDialAnswerTime 指定的时间内应答
22	SS1OUT_NOBWDACK	SS1	去话呼叫失败：等待对端交换机的“占用证实”信号超时。
23	SS1OUT_DIALING_BWD_HANGUP	SS1	去话呼叫失败：对端交换机取消呼叫
24	SS1OUT_BWD_A5	SS1	去话呼叫失败：收到对端交换机的 A5 信号（空号）
25	SS1OUT_BWD_KB5	SS1	去话呼叫失败：收到对端交换机的 KB=5（空号）
26	SS1OUT_BWD_KB2	SS1	去话呼叫失败：收到对端交换机的 KB=2（被叫用户“市忙”）
27	SS1OUT_BWD_KB3	SS1	去话呼叫失败：收到对端交换机的 KB=3（被叫用户“长忙”）
28	SS1OUT_BWD_A4	SS1	去话呼叫失败：收到对端交换机的 A4 信号（机键拥塞）
29	SS1OUT_BWD_KB4	SS1	去话呼叫失败：收到对端交换机的 KB=4 信号（机键拥塞）
30	SS1OUT_TIMEOUT_BWD_A	SS1	去话呼叫失败：等待对端交换机的后向 A 组信号超时
31	SS1OUT_TIMEOUT_BWD_A_STOP	SS1	去话呼叫失败：等待对端交换机停发后向 A 组信号超时
32	SS1OUT_TIMEOUT_BWD_KB	SS1	去话呼叫失败：等待对端交换机的 KB 信号超时
33	SS1OUT_TIMEOUT_BWD_KB_STOP	SS1	去话呼叫失败：等待对端交换机停发 KB 信号超时
34	SS1OUT_TIMEOUT_CALLERID_BWD_A1	SS1	去话呼叫失败：向对端交换机发送主叫号码时，等待对端交换机后向 A 组信号超时
35	SS1OUT_TIMEOUT_CALLERID_BWD_A1_STOP	SS1	去话呼叫失败：向对端交换机发送主叫号码时，等待对端交换机停发后向 A 组信号超时，自动拨号失败
36	SS1OUT_UNDEFINED_CALLERID_BWD_A	SS1	去话呼叫失败：向对端交换机发送主叫号码时，收到未定义的后向 A 组信号
37	SS1OUT_UNDEFINED_BWD_A		去话呼叫失败：收到未定义的后向 A 组信号
38	SS1OUT_UNDEFINED_BWD_KB		去话呼叫失败：收到未定义的 KB 信号
41	ISDN_CALLOVER	ISDN	呼叫结束，对方先挂机
42	ISDN_WAIT_RELEASE	ISDN	收到对方的“拆线”消息，等待释放本端链路
43	ISDN_HANGING	ISDN	本端先挂机，正在拆线
44	ISDN_RELEASING	ISDN	正在释放呼叫
45	ISDN_UNALLOCATED_NUMBER	ISDN	未分配的号码
46	ISDN_NETWORK_BUSY	ISDN	网络忙
47	ISDN_CIRCUIT_NOT_AVAILABLE	ISDN	指定的电路不可用
48	PEND_CalleeHangupOnWaitRemotePickUp	TUP	去话呼叫失败：等待被叫用户摘机时，收到对端交换机的拆线消息
49	ISUP_HardCircuitBlock	ISUP	收到对端交换机的硬件闭塞消息
50	ISUP_RemoteSuspend	ISUP	T6 定时器溢出。有关T6 定时器的更多信息请参见第 1 章中“ISUP通道的状态转移”部分内容。

51	PEND_RcvHGBOrSGB	TUP	收到对端交换机闭塞消息（SGB/HGB）
52	ISDN_NO_ANSWER	ISDN	无应答
53	ISDN_CALL_REJ	ISDN	呼叫拒绝
54	PEND_RemoteHangupOnRinging	ISUP / TUP	来话呼叫：通道处于“振铃”状态时，对端交换机取消了呼叫
55	ISDN_NO_ROUTE	ISDN	无法路由，原因可能为“移动电话不在服务区”
56	ISDN_NO_ROUTE_TO_DEST	ISDN	无法路由，原因可能为“移动电话关机”
57	EM_USER_BUSY	E/M	用户忙
58	EM_CH_ERROR	E/M	通道故障
59	EM_LOCAL_HANGUP	E/M	本地先挂机
60	EM_LOCAL_NOANSWER	E/M	本地无应答
61	EM_REMOTE_HANGUP	E/M	对端先挂机
62	EM_REMOTE_NOANSWER	E/M	对端无应答
63	PEND_RemoteHangupOnSuspend	ISUP	“暂停”状态时，对端用户挂机
64	PEND_CalleeHangupOnSuspend	ISUP	“暂停”状态时，被叫用户挂机
65	ISDN_NORMAL_UNSPEC	ISDN	呼叫正常结束
其它	保留		

功能描述：

获取通道迁移到 S_CALL_PENDING 状态的具体原因。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.3.5 通道摘机函数（CTI 系列）

2.3.5.1 SsmPickup

2.3.5.2 SsmPickupANX

在通道上执行摘机操作。SsmPickupANX 除了具备 SsmPickup 的全部功能外，还可以在应答消息中指示被叫用户的计费情况，但只适用于 TUP 通道和 ISUP 通道。

函数原型：

int SsmPickup(int ch)

int SsmPickupANX(int ch, int nANX)

参数说明：

ch	通道编号
nANX	=0：计费未说明 =1：计费 =2：免费

返回值：

-1	操作失败
0	操作成功

功能描述：

在通道上执行摘机操作。

✧ **ISUP 通道：**

对于ISUP通道，应用程序调用本函数时，如果通道处于“振铃”状态，驱动程序会根据配置项DefaultCalledPickupMsg的设置向对端交换机发送CON（地址全并摘机）或ANM（应答）消息，消息中的后向呼叫表示语字段可以通过配置项DefaultBackwardCallInd进行设置；如果通道处于“空闲”或“锁定”状态，驱动程序不会向对端交换机发送任何消息。

✧ **ISDN 通道：**

对于 ISDN 通道，应用程序调用本函数时，如果通道处于“振铃”状态，驱动程序会向对端交换机发送 CONNECT 消息，同时启动定时器 T313，然后将通道迁移到 S_ISDN_IN_WAIT_TALK 状态；如果通道处于“空闲”或“锁定”状态，驱动程序不会向对端交换机发送任何消息。

✧ **模拟中继线通道：**

对于模拟中继线通道的来话呼叫，应用程序调用本函数时，如果通道处于S_CALL_RINGING状态，并且铃流信号的电平为高电平，为防止线路上的铃流电压对板卡造成损害，驱动程序不会马上向硬件电路发布摘机命令。只有当铃流信号转变为低电平后，驱动程序才会真正执行摘机命令，并向应用程序抛出E_SYS_ActualPickup事件。函数SsmCheckActualPickup也可以用来查询摘机命令是否执行完毕。

✧ **IP 通道：**

对处于“空闲”状态的IP通道调用本函数，通道将处于“呼出保留”状态。在“呼出保留”状态下，用户可调用SsmHangup使通道回到空闲，也可调用拨号函数呼出。如果用户在 60s内未进行任何操作，通道将自动回到空闲。对处于“振铃”状态的IP通道调用本函数，其意义相当于接收来电。其他状态下调用本函数，返回错误。

注意事项：

- SsmPickup 适用于 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、模拟中继线通道、IP 通道，SsmPickupANX 只适用于 TUP 通道和 ISUP 通道；
- 对于模拟中继线通道，应用程序调用本函数后，驱动程序会自动清零铃流信号检测器，清零并启动信号音检测器。

相关信息：

驱动程序版本要求	SsmPickup 要求 SynCTI Ver. 3.0 或更高 SsmPickupANX 要求 SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmHangup，SsmHangupEx

2.3.6 通道挂机函数（CTI 系列）

2.3.6.1 SsmHangup

2.3.6.2 SsmHangupEx

向通道发布挂机命令，SsmHangupEx 还可以向驱动程序传递挂机原因等信息，但只适用于 ISUP 通道。

函数原型：

int SsmHangup(int ch)

int SsmHangupEx(int ch, UCHAR ucCauseVal)

参数说明：

ch	通道编号
ucCauseVal	挂机的释放原因，具体值请参考 ISUP 相关协议说明

返回值：

-1	初始化失败，失败原因可以通过SsmGetLastErrMsg函数获得
0	初始化成功

功能描述：

向通道发布挂机命令。驱动程序执行的操作取决于通道的类型。

- **ISUP 通道**

本函数会引发驱动程序向对端交换机发送 REL（释放）消息，可以用来拒绝来话呼叫，也可以用来取消去话呼叫。如果应用程序调用的是 SsmHangupEx，REL 消息中的释放原因由参数 ucCauseVal 决定；如果调用的是 SsmHangup，释放原因会因通道的状态而异：

通道状态	REL 消息中的释放原因
S_CALL_PICKUPED	不发送 REL 消息
S_ISUP_WaitDialAnswer S_CALL_WAIT_REMOTE_PICKUP S_CALL_TALKING S_ISUP_WaitSetCallerID	C_ISUP_REL_NORMAL_REL
S_ISUP_Suspend	C_ISUP_REL_DENY
S_ISUP_WaitSAM S_ISUP_WaitINF S_CALL_RINGING S_CALL_PENDING（挂起原因为 PEND_WaitBckStpMsg）	由配置项DefaultHangupRELInd或函数SsmSetIsupFlag（携带ISUP_REL_DENY_SetToOther参数）的设定值决定

注：上表中使用的宏在头文件 shpa3api.h 中声明。

- **TUP 通道**

如果通道处于S_CALL_RINGING状态，调用本函数表明应用程序拒绝来话呼叫，驱动程序会向对端交换机发送呼叫拒绝消息。呼叫拒绝消息可以是CBK或CFL消息，具体使用哪一个，由配置项HangupRingSendCBK决定。

- **ISDN 通道**

对于 ISDN 通道，调用本函数后：

- ✧ 如果通道处于 S_CALL_PENDING 状态，驱动程序会向对端交换机发送“RELEASE”消息；
- ✧ 如果通道处于 S_ISDN_OUT_PLS_APPEND_NO 状态，本次呼叫将被挂断，驱动程序会向对端交换机发送 “DISCONNECT” 消息，消息中的挂机原因字段会被设置为“号码未分配”；
- ✧ 如通道处于 S_CALL_TALKING 状态，表明本端先挂机，驱动程序会向对端交换机发送“DISCONNECT” 消息，消息中的挂机原因会被设置为“呼叫正常结束”；
- ✧ 如果通道处于 S_CALL_RINGING 状态，表明应用程序拒绝本次来话呼叫，驱动程序会向对端交换机发送 “DISCONNECT” 消息，消息中的挂机原因将被设置为“呼叫拒绝”。

- **IP 通道**

对于 IP 通道，调用本函数后：

- ✧ 对处于“呼出保留”状态的 IP 通道调用本函数，通道将迁移到“空闲”状态；
- ✧ 对处于非空闲状态的的 IP 通道调用本函数，IP 状态机将根据当前的状态，触发不同的协议命令，并使 IP 会话结束。

注意事项：

- SsmHangupEx 函数只适用于 ISUP 通道；
- 对于ISDN通道，应用程序可以在调用本函数前先调用函数SsmISDNSetHangupRzn设置挂机原因。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高 SsmHangupEx： SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPickup，SsmPickupANX

2.3.7 设置通道的呼叫方向

2.3.7.1 SsmSetAutoCallDirection

设置通道的呼叫方向。

函数原型：

int SsmSetAutoCallDirection(int ch, BOOL bEnAutoCall, int nDirection)

参数说明：

ch	通道编号
bEnAutoCall	是否由驱动程序自动进行呼叫处理。 =TRUE： 是； =FALSE：否。此时，nDirection 将被忽略。
nDirection	呼叫方向，仅当 bEnAutoCall 设置为 TRUE 时有效。取值范围： =0：只能进行来话呼叫； =1：只能进行去话呼叫； =2：既可以进行来话呼叫，也可以进行去话呼叫。

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
≥0	调用成功

功能描述：

设置通道的呼叫方向。

注意事项：

- 本函数只支持 SHD 系列板卡的 SS1 通道；
- 只有在通道处于 S_CALL_STANDBY 状态时才能调用本函数；
- 本函数对SS7、ISDN中继通道不起作用。如果需要对SS7 和ISDN通道设置呼叫方向，可以通过阻塞函数来设置，具体参考函数SsmBlockLocalCh。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetChState，SsmGetAutoCallDirection

2.3.7.2 SsmGetAutoCallDirection

获取通道的呼叫方向。

函数原型：

int SsmGetAutoCallDirection(int ch, int* pnDirection)

参数说明：

ch	通道编号
pnDirection	返回呼叫方向，仅当 bEnAutoCall 设置为 TRUE 时有效。取值范围： =0：只能进行来话呼叫； =1：只能进行去话呼叫； =2：既可以进行来话呼叫，也可以进行去话呼叫。

返回值：

-1	调用失败
0	本通道不是由驱动程序自动进行呼叫处理
1	本通道由驱动程序自动进行呼叫处理

功能描述：

获取通道的呼叫方向。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetAutoCallDirection

2.3.8 获取释放原因

2.3.8.1 SsmGetReleaseReason

获取对端交换机发送的释放消息中的释放原因值。

函数原型：

WORD SsmGetReleaseReason(int ch)

参数说明：

ch	通道编号

返回值：

0	原因未指明
>0	释放消息中的原因值，具体的消息类型及含义与通道类型有关： ➢ ISUP 通道：返回 REL 消息中的释放原因，具体取值及含义为： 0x01：空号 0x10：正常拆线 0x11：用户忙 0x12：无应答 0x15：拒绝 0x1c：地址不全 0x1f：正常 0x2a：交换设备拥塞 0x14：用户缺席 0x16：号码改变 ➢ ISDN 通道：返回 ISDN 消息 ISDN 通道释放原因值表字段，具体取值及含义请参见附录 1 ISDN 通道释放原因值表。

功能描述：

获取对端交换机发送的释放消息中的释放原因值。

注意事项：

- 在开始处理来话呼叫前（即当驱动程序从对端交换机收到一条呼叫建立消息时），驱动程序都会将释放原因值设置为 0；
- 本函数只适用于 ISDN 通道（包括网络侧和用户侧）和 ISUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.3.9 TUP 通道专用函数

2.3.9.1 SsmSetCalleeHoldFlag

设置 TUP 通道的“锁定主叫用户”功能。

函数原型：

int SsmSetCalleeHoldFlag(int ch, BOOL bFlag)

参数说明：

ch	通道编号
bFlag	=TRUE： 开启 =FALSE：关闭

返回值：

-1	操作失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	操作成功

功能描述：

设置TUP通道的“锁定主叫用户”功能。更多信息请参见第 1 章中“锁定主叫用户功能”部分内容。

注意事项：

- 本函数只适用于 SHD 系列板卡的 TUP 通道；
- 只用当通道处于 S_CALL_TALKING 状态时，才可以调用本函数；
- 每次来话呼叫时，驱动程序在将通道迁移到 S_CALL_TALKING 状态前，会自动将“锁定主叫用户”功能设置为关闭状态。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.3.9.2 SsmGetTupFlag

获取 TUP 消息中的特定参数或字段值。

函数原型：

int SsmGetTupFlag(int ch,int nType,DWORD *pdwParaValue)

参数说明：

ch	通道编号
nType	选择参数或字段类型。取值范围： =1(Tup_ANX)：去话呼叫时，当通道处于 S_CALL_WAIT_REMOTE_PICKUP 状态时，如果被叫用户接受此呼叫，对端交换机会向本端发送应答信号(Answer Signal)，驱动程序会将通道迁移到 S_CALL_TALKING 状态。此时，参数 pdwParaValue 可以获取具体的应答消息类型。
pdwParaValue	返回参数或字段的值。具体含义与 nType 有关： nType=Tup_ANX： 0x06：ANU 消息（计费未说明，Unqualified） 0x16：ANC 消息（计费，Charge） 0x26：ANN 消息（免费，No Charge）

返回值：

1	成功获取消息。
-1	调用失败，失败原因可以从SsmGetLastErrMsg函数中获取

功能描述：

获取 TUP 消息中的特定参数或字段值。

注意事项：

- 本函数只适用于 TUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.9 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：无

2.3.9.3 SsmSetTupParameter

设置呼出通道的 Tup 消息可选参数。

函数原型：

int SsmSetTupParameter (int nBCh, UCHAR ucMsgTypeCode, UCHAR ucParamTypeCode, WORD wLength, PUCHAR pucContent);

参数说明：

nBCh	通道编号
ucMsgTypeCode	TUP 消息类型
ucParamTypeCode	参数类型
wLength	字节数
pucContent	参数内容的缓冲区指针

返回值：

-1	参数设置失败
0	设置成功

功能描述：

设置呼出通道的 Tup 消息可选参数。目前只对 IAI 中原被叫地址性质表示语有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：无

2.3.10 ISUP 通道专用函数

2.3.10.1 SsmSetISUPCAT

设置 ISUP 协议中 IAM 消息中的主叫用户类别。

函数原型：

int SsmSetISUPCAT(int nch, UCHAR ucCallerCAT)

参数说明：

nch	通道编号
ucCallerCAT	十六进制的主叫用户类别

返回值：0

功能描述：

设置ISUP协议中IAM消息中的主叫用户类别。本函数设置的参数也可以通过配置项DefaultIAM_CAT进行设置。

注意事项：

- 本函数应该在调用SsmAutoDial或SsmAutoDialEx前调用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.3.10.2 SsmSetIsupParameter

设置本端发送给对端交换机的 isup 消息中的可选字段。

函数原型：

int SsmSetIsupParameter(int ch, UCHAR ucMsuType, UCHAR ucParamType, WORD wLength, PUCHAR pucContent)

参数说明：

ch	通道编号
ucMsuType	消息类型。取值范围： 0x01：IAM 消息 0X09：ANM 消息 其它：保留
ucParamType	参数类型。具体取值与 ucMsuType 有关： ucMsuType= IAM 消息：可以设置可选字段 ucMsuType= ANM 消息：可以设置除了下列可选字段以外的其它可选字段： ✧ 后向呼叫指示 每个可选字段的参数类型请参见 ISUP 协议的相关说明。 ucMsuType=其它：保留
wLength	参数的长度，单位为字节。如果 ucMsuType=IAM 消息，将 wLength 设置为 0 表示 IAM 消息中不再包含 ucParamType 指定的可选字段。
pucContent	参数的内容。如果 ucMsuType=IAM，将 pucContent 设置为 NULL 表示 IAM 消息中不再包含 ucParamType 指定的可选字段。

返回值：

-1	调用失败。

0	调用成功。

功能描述：

设置本端发送给对端交换机的 isup 消息中的可选字段。

注意事项：

- 如果ucMsuType设置为IAM消息，本函数应该在SsmAutoDial之前调用。参数的值一旦设置后，会被驱动程序保存在内部缓冲区中，并会在后续的IAM消息中继续使用。对于IAM消息中的用户到用户信息字段，也可以通过配置项Usr2UsrInfo进行设置。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmGetIsupParameter

代码范例：

在 IAM 消息中增加改发信息字段。改发信息字段的参数类型为 0x13，共包含 2 个字节，改发信息参数字段中的第 1 个 8 位位组（记为 HGFEDCBA，A 为最低 bit 位）的含义为：

比特	含义	取值	描述
CBA	改发表示语	000	无改发
		001	呼叫重新选路
		010	呼叫重新选路，所有的改发信息提供限制
		011	呼叫变更
		100	呼叫变更，所有的改发信息提供限制
		110	呼叫变更，改发号码提供限制
		其它	保留
D	备用		
HGFE	原改发原因	0000	未知 / 不可用
		0001	用户忙
		0010	无应答
		0011	无条件
		其它	保留

第 2 个 8 位位组（记为 PONMLKJI，I 为最低 bit 位）的含义为：

比特	含义	取值	描述
KJI	改发计数器	1~5	呼叫经过改发的次数用 1 和 5 之间的二进制数表示
L	备用		应设置为 0
PONM	改发原因	0000	未知 / 不可用
		0001	用户忙
		0010	无应答
		0011	无条件
		0110	移动用户不可及
		其它	保留

如果设置改发信息为用户无应答前转，则相关的代码如下：

```
char IsupParamRI[2];
IsupParamRI[0] = 0x03;          //设置改发信息内容
IsupParamRI[1] = 0x21;
SsmSetIsupParameter(ch, 0x01, 0x13, 2, IsupParamRI);
SsmAutoDial(ch,…);
```

2.3.10.3 SsmGetIsupParameter

获取对端交换机发送给本端的 isup 消息中的可选字段。

函数原型：

int SsmGetIsupParameter(int nBCh, UCHAR ucMsuType, UCHAR ucParamType, PUCHAR pucContent, WORD wBufSize, LPWORD lpNumberOfBytesWritten)

参数说明：

nBCh	通道编号
ucMsuType	消息类型。取值范围： 0x01：IAM 消息 其它：保留
ucParamType	可选字段的类型。具体取值与 ucMsuType 有关，更多信息请参见 ISUP 协议的相关说明。
pucContent	保存可选字段内容的缓冲区指针，由应用程序分配存储空间，缓冲区长度的建议值为 272。**注意：**返回的信息内容中不包括可选字段的类型编码（1 字节）和长度（1 字节）。
wBufSize	pucContent 的长度，单位为字节
lpNumberOfBytesWritten	返回驱动程序写入到 pucContent 中的数据的实际长度，单位为字节

返回值：

-1	调用失败。
0	调用成功。

功能描述：

获取对端交换机发送给本端的 isup 消息中的可选字段。

注意事项：

- 本函数目前只支持 IAM 消息。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmSetIsupParameter

2.3.10.4 SsmSetIsupFlag

ISUP 中用户部分参数设置。

函数原型：

int SsmSetIsupFlag(int ch, int nType, DWORD dwValue, PVOID pV)

参数说明：

Ch	通道号
nType	选择参数的类别。可取的值为： ISUP_PhoNumParam：设置 IAM 消息中被叫用户号码字段的被叫参数，此时参数 dwValue 有效。 ISUP_CallerParam：设置主叫用户号码字段的主叫参数，此时参数 dwValue 有效。 ISUP_REL_DENY_SetToOther：当应用程序调用SsmHangup函数而引发驱动程序向对端交换机发送REL消息时，本参数可以设置REL消息中携带的释放原因值。 ISUP_PhoNumREL：向对端交换机发送包含号码改发信息的拆线消息（REL），此时参数 pV 有效。 DefaultIAM_OriginalCalleeParam：原被叫参数。
dwValue	设置参数的值，与 nType 的取值有关： **nType 取值** / **dwValue 的取值及含义** ISUP_PhoNumParam：请参见配置项DefaultIAM_CalleeParam的说明 ISUP_CallerParam：请参见配置项DefaultIAM_CallerParam的说明 ISUP_REL_DENY_SetToOther：请参见配置项DefaultHangupRELInd的说明 DefaultIAM_OriginalCalleeParam：原被叫参数的值： 备　用：0x1000 用户号码：0x1001 未　知：0x1002

	国内号码：0x1003 国际号码：0x1004 ISUP_PhoNumREL　保留未用
pV	pV 的含义与 nType 的取值有关： ➢ nType=ISUP_PhoNumREL： pV 是指向 ISUP_RIREL 结构对象的指针，用于定制 REL 消息中的参数。ISUP_RIREL 结构在 Shpa3api.h 中声明如下： typedef struct tag_ISUP_RIREL{ WORD **wRIMsg**; WORD **wRIPhoNumPara**; WORD **wPhoNumLen**; UCHAR **ucRIPhoNum**[20]; }ISUP_RIREL,*PISUP_RIREL; 上述结构中的参数含义为： **wRIMsg**：“改发信息”字段的比特 PONMLKJIHGFEDCBA； **wRIPhoNumPara**：“改发号码”字段的比特 Bit15…Bit0； **wPhoNumLen**：ucRIPhoNum 中字符的个数； **ucRIPhoNum**：改发的电话号码字符串。 ➢ nType=其它：保留未用。

注：表中用到的常量均在头文件 Shpa3api.h 中声明。

返回值：

-1	失败
1	成功设置

功能描述：

设置 ISUP 中用户部分参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIsupFlag

2.3.10.5 SsmGetIsupFlag

获取 ISUP 中用户部分参数设置。

函数原型：

int　SsmGetIsupFlag(int ch, int nType, DWORD *pd)

参数说明：

ch	通道号
nType	选择参数类型，详细内容请参见函数SsmSetIsupFlag的说明。
*pd	参数的值，详细内容请参见函数SsmSetIsupFlag的说明。

返回值：

-1	失败
1	调用成功

功能描述：

获得 ISUP 中用户部分参数的设置值。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIsupFlag

2.3.10.6 SsmGetIsupUPPara

获取保存在驱动程序内部的 ISUP 原始消息。

函数原型：

int SsmGetIsupUPPara(int nBCh, WORD wMsuType, LPWORD pwMsuSize, PUCHAR pucRawMsu)

参数说明：

nBCh	通道编号
wMsuType	消息类型 **取值** / **shpa3api.h 中的宏** / **pucContent 参数对应的消息** 0x01 / C_ISUP_IAM / 初始地址消息（IAM） 0x06 / C_ISUP_ACM / 地址全消息（ACM）
pwMsuSize	返回消息的实际长度，单位为字节。
pucRawMsu	存放 ISUP 消息的缓冲区指针。缓冲区由应用程序分配，长度必须大于或等于 281 字节。

返回值：

1	调用成功
-1	错误的通道编号
-2	调用失败，参数 wMsuType 越界

功能描述：

获取保存在驱动程序内部的ISUP原始消息。更多信息请参见第 1 章中“ISUP通道的状态转移”部分内容。

注意事项：

- 本函数只支持 ISUP 通道；

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIsupUPPara

2.3.10.7 SsmSetIsupUPPara

向驱动程序提交一个自行构建的 ISUP 消息。

函数原型：

int SsmSetIsupUPPara(int nBCh, WORD wMsuType, LPWORD pwMsuSize, PUCHAR pucRawMsu)

参数说明：

nBCh	通道号
wMsuType	消息类型 **取值** / **shpa3api.h 中的宏** / **pucContent 参数对应的消息** 0x01 / C_ISUP_IAM / 初始地址消息（IAM） 0x06 / C_ISUP_ACM / 地址全消息（ACM）

pwMsuSize	ISUP 消息的长度，包括 SIO 字段，单位为字节，最大不能超过 281 个字节
pucRawMsu	消息数据的首地址指针

返回值：

1	成功
-1	错误的通道编号
-2	调用失败，参数 wMsuType 越界
-3	校验失败，消息内容不符合 ISUP 协议规则

功能描述：

向驱动程序提交一个自行构建的 ISUP 消息。

注意事项：

- 本函数只支持 ISUP 通道；
- 如果wMsuType为C_ISUP_IAM，本函数必须在SsmAutoDial或SsmAutoDialEx前调用；
- 如果wMsuType为C_ISUP_ACM，应用程序提交的ACM消息只有在通道处于“挂起”状态，并且挂起原因为PEND_WaitBckStpMsg时才可以被发送给对端交换机。更多信息请参见第 1 章中“ISUP通道的状态转移”部分内容；
- 驱动程序在发送应用程序提交的ISUP消息时，会自动覆盖消息中的SIO、DPC、OPC、SLC、CIC字段。如果wMsuType为C_ISUP_IAM，驱动程序还会用SsmAutoDial或SsmAutoDialEx中的被叫号码覆盖pucRawMsu 中的相应字段。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmAutoDialEx，SsmGetIsupUPPara，SsmSendIsupMsg

2.3.10.8 SsmSendIsupMsg

向对端交换机发送一条 ISUP 消息。

函数原型：

int SsmSendIsupMsg(int nBCh, WORD wMsuType)

参数说明：

nBCh	通道号
wMsuType	消息类型。 取值 | shpa3api.h 中的宏 | 消息类型 0x06 | C_ISUP_ACM | 地址全消息（ACM）

返回值：

1	成功
-1	错误的通道编号
-2	调用失败，参数 wMsuType 越界
-3	发送消息失败

功能描述：

向对端交换机发送一条ISUP消息。ISUP消息必须预先通过调用SsmSetIsupUPPara函数提交给驱动程序。

注意事项：

- 本函数只支持 ISUP 通道；
- 只有在通道处于 S_CALL_PENDING 状态，并且挂起原因为 PEND_WaitBckStpMsg 时，才允许调用本函

数。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIsupUPPara

2.3.10.9 SsmGetRedirectionInfReason

来话呼叫时，从收到的 IAM 消息里取出转呼的原因值。

函数原型：

int SsmGetRedirectionInfReason(int ch)

参数说明：

ch	通道号

返回值：

-1	失败
≥0	转呼的原因值： 0：不知/不可获得 1：用户忙 2：无应答 3：无条件 4：通知时偏转 5：偏转立即响应 6：移动用户不可达到

功能描述：

来话呼叫时，从收到的 IAM 消息里取出转呼的原因值。

注意事项：

- 本函数只支持 ISUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.6 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIsupUPPara

2.3.10.10 SsmIsHaveCpg

2.3.10.11 SsmGetCpg

SsmIsHaveCpg 查询 ACM 消息后是否跟随了 CPG 消息，如果有，SsmGetCpg 用于取出保存在驱动程序内部的原始 CPG 消息。

函数原型：

int SsmIsHaveCpg(int ch)

int SsmGetCpg(int ch, char* pszContent, int* piLength)

参数说明：

ch	通道编号
pszContent	返回 CPG 消息的内容。有关 CPG 消息的详细信息请参见 ISUP 协议的相关文档。
piLength	返回 CPG 消息的长度

返回值：

-1	操作失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	SsmIsHaveCpg：没有 CPG 消息 SsmGetCpg：调用成功
1	SsmIsHaveCpg：有 CPG 消息。只有当 SsmIsHaveCpg 返回 1 时，调用 SsmGetCpg 才有意义。

功能描述：

去话呼叫时，对端交换机在发送了 ACM 消息后，可能会继续发送 CPG 消息（呼叫进展）。驱动程序收到此 CPG 消息后，会将其保存在内部缓冲区中。本函数用于查询 ACM 消息后是否跟随了 CPG 消息。

注意事项：

- 只有当通道处于 S_CALL_WAIT_REMOTE_PICKUP 状态时，才可以调用本函数；
- 每当发生下列事件时，驱动程序会自动清除保存在内部缓冲区中的 CPG 消息：
  - 通道迁移到 S_CALL_STANDBY 状态；
  - 收到对端交换机的 ACM 消息。
- 本函数只支持 ISUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：E_CHG_ChState

2.3.10.12 SsmIsupGetUsr

检查并获取 USR 消息。

函数原型：

BOOL SsmIsupGetUsr(int *ch, PUCHAR pucData, PUCHAR ucLen)

参数说明：

ch	收到消息的通道号的指针
pucData	用于存放收到的 USR 消息中“用户至用户信息”的数据的缓冲区，需要 130 字节
ucLen	pucData 的长度

返回值：

0	取到新的数据
其他	未取到新的数据

功能描述：

检查并获取 USR 消息。执行本函数后，新的消息将从系统中清除。

注意事项：无。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关事件： E_RCV_Ss7IsupUtuinf

2.3.10.13 SsmIsupSendUsr

发送 USR 消息。

函数原型：

BOOL SsmIsupSendUsr(int ch, PUCHAR pucData, UCHAR ucLen)

参数说明：

ch	通道号
pucData	作为 USR 消息中的“用户至用户信息”发送的数据
ucLen	pucData 的长度，协议中规定有效值为 2~130

返回值：

0	调用成功
其他	调用失败

功能描述：

发送 USR 消息。

注意事项：无。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：

2.3.11 ISDN 通道专用函数

本节提供的函数专用于 ISDN 通道的呼叫。

2.3.11.1 设置 Setup 消息的参数

2.3.11.1.1 SsmISDNSetDialSubAddr

去话呼叫时，设置 setup 消息中的被叫方子地址。

函数原型：

int SsmISDNSetDialSubAddr(int ch, LPSTR lpSubAddress)

参数说明：

ch	通道编号
lpSubAddress	被叫的子地址，由字符‘0’~‘9’组成，最大长度为 20 个字节，由应用程序分配存储空间

返回值：

0	调用成功
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

去话呼叫时，设置 setup 消息中的被叫方子地址。

注意事项：

- 本函数必须在SsmAutoDial函数之前调用。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial

2.3.11.1.2SsmISDNSetTxSubAddr

去话呼叫时，设置 setup 消息中的主叫子地址。

函数原型：

int SsmISDNSetTxSubAddr(int ch, LPSTR lpSubAddress)

参数说明：

ch	通道编号
lpSubAddress	主叫用户子地址，由字符‘0’~‘9’组成，最大长度为 20 个字节，由应用程序分配存储空间。

返回值：

0	设置成功
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

去话呼叫时，设置 setup 消息中的主叫用户子地址信息。

注意事项：

- 本函数必须在SsmAutoDial函数之前调用。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAutoDial，SsmISDNGetTxCallerSubAddr

2.3.11.2 获取来话呼叫信息

2.3.11.2.1获取子地址信息

2.3.11.2.1.1 SsmISDNGetSubAddr

获取被叫用户子地址信息。

函数原型：

int SsmISDNGetSubAddr(int ch, LPSTR lpSubAddress)

参数说明：

ch	通道编号
lpSubAddress	存放被叫用户子地址信息的缓冲区，由应用程序分配存储空间，长度不能少于 21 字节

返回值：

≥0	lpSubAddress 中字符的实际长度，单位为字节
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

来话呼叫时，来自对端交换机的消息中包含了被叫用户子地址信息。本函数获取从消息中提取出来的 ASCII 字符形式的被叫用户子地址信息。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmISDNGetCallerSubAddr

2.3.11.2.1.2 SsmISDNGetCallerSubAddr

获取主叫用户子地址信息。

函数原型：

int SsmISDNGetCallerSubAddr(int ch, LPSTR lpSubAddress)

参数说明：

ch	通道编号
lpSubAddress	存放主叫用户子地址信息的缓冲区，由应用程序分配存储空间，长度不能少于 21 字节

返回值：

≥0	主叫子地址长度
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

来话呼叫时，来自对端交换机的消息中包含了主叫用户子地址信息。本函数获取从消息中提取出来的 ASCII 字符形式的主叫用户子地址信息。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmISDNGetSubAddr

2.3.11.2.2 获取用户自定义主叫号码

2.3.11.2.2.1 SsmGetUserCallerId

获取用户自定义主叫号码。

函数原型：

int SsmGetUserCallerId(int ch, LPSTR szCallerId)

参数说明：

ch	通道编号
szCallerId	放置呼入时用户自定义主叫号码的缓冲区。最大长度 21 字节

返回值：

-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因
≥0	自定义主叫号码长度

功能描述：

获取用户自定义主叫号码。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.3.11.3 设置挂机原因

2.3.11.3.1 SsmISDNSetHangupRzn

设置挂机原因。

函数原型：

int SsmISDNSetHangupRzn(int ch, int nReason)

参数说明：

ch	通道编号
nReason	挂机原因，取值范围： 0：正常挂机 1：用户忙 2：空号 3：拒绝本次呼叫

返回值：

0	调用成功
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

设定挂机原因。

注意事项：

- 本函数必须在SsmHangup函数之前调用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPendingReason

2.3.11.4 其它函数

2.3.11.4.1 SsmISDNGetDisplayMsg

获取 ISDN 通道的显示信息。

函数原型：

int SsmISDNGetDisplayMsg(int ch, LPSTR lpDispMsg)

参数说明：

ch	通道编号
lpDispMsg	存放显示信息的缓冲区，最大长度为 100 个字节

返回值：

≥0	显示信息长度
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

获取 ISDN 通道的显示信息。显示信息为 ASCII 形式的字符串，通常由网络侧发送给用户侧，可能包含问候语或其他的附加信息。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：无

2.3.11.4.2 SsmISDNGetTxCallerSubAddr

获取本端设置的主叫方子地址信息。

函数原型：

int SsmISDNGetTxCallerSubAddr(int ch, LPSTR lpSubAddress)

参数说明：

ch	通道编号
lpSubAddress	放置呼出时主叫子地址信息的缓冲区。最大长度 21 字节

返回值：

≥0	主叫子地址长度
-1	调用失败，函数SsmGetLastErrMsg可以获取失败原因

功能描述：

获取本端设置的主叫方子地址信息。

注意事项：

- 本函数只适用于去话呼叫。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmISDNGetSubAddr

2.4 有关信号音发生器的函数（CTI 系列）

2.4.1 设置信号音发生器的工作参数

2.4.1.1 SsmSetTxTonePara

本函数设置信号音发生器产生的信号音的频率和音量。

函数原型：

int SsmSetTxTonePara(int ch, int nFreq1, int nVolume1, int nFreq2, int nVolume2)

参数说明：

ch	通道编号
nFreq1	设置第 1 音频的频率，单位为 Hz，取值范围：300~3400，缺省值为 450
nVolume1	设置第 1 音频的音量，取值范围：-7~+6。大于 0 表示音量提升，小于 0 表示音量衰减，-7 表示关闭第 1 音频。缺省值为 0。取值乘以 3 即为分贝值。
nFreq2	设置第 2 音频的频率，单位为 Hz，取值范围：300~3400，缺省值为 0，表示不使用本频率，只发送单音频信号音。
nVolume2	设置第 2 音频的音量，取值范围：-7~+6。大于 0 表示音量提升，小于 0 表示音量衰减，-7 表示关闭第 2 音频。缺省值为-7（不使用第 2 音频）。取值乘以 3 即为分贝值。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

本函数设置信号音发生器产生的信号音的频率和音量。

注意事项：

- 配置项“DefaultSendToneFrequence和DefaultSendToneVolume可以实现相同功能，缺省的信号音发生器为 450Hz单音频信号音。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendTone

2.4.1.2 SsmQueryOpSendTone

查询通道是否具有信号音发生器。

函数原型：

int SsmQueryOpSendTone (int ch)

参数说明：

Ch	通道编号

返回值：

-1	调用失败
0	没有
1	有

功能描述：

查询通道是否具有信号音发生器。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.4.1.3 SsmGetTxTonePara

获取信号音发生器中的频率参数和音量。

函数原型：

int SsmGetTxTonePara(int ch, int* nFreq1, int* nVolume1, int* nFreq2, int* nVolume2)

参数说明：

ch	通道号
nFreq1	返回信号音参数中第 1 音频的频率，单位：Hz
nVolume1	返回信号音参数中第 1 音频的音量，取值范围：-7~+6，大于 0 表示音量提升，小于 0 表示音量衰减，-7 表示不使用该频率。取值乘以 3 即为分贝值。
nFreq2	返回信号音参数中第 2 音频的频率，单位为 Hz
nVolume2	返回信号音参数中第 2 音频的音量，取值范围：-7~+6，大于 0 表示音量提升，小于 0 表示音量衰减，-7 表示不使用该频率。取值乘以 3 即为分贝值。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

取得本通道上送信号音时的频率和音量。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetTxTonePara

2.4.2 发送信号音

2.4.2.1 SsmSendTone

2.4.2.2 SsmSendToneEx

启动信号音发生器，在线路上产生信号音。SsmSendTone 用于发送驱动程序预定义类型的信号音，SsmSendToneEx 可以指定信号音的高电平保持时间和低电平保持时间。

函数原型：

int SsmSendTone(int ch, int nToneType)

int SsmSendToneEx(int ch, DWORD dwOnTime, DWORD dwOffTime)

参数说明：

ch	通道编号
nToneType	信号音类型，取值包括： 0：拨号音 1：忙音 2：回铃音 3：催挂音
dwOnTime	信号音高电平持续时间，单位为毫秒，必须为 8 的整数倍。
dwOffTime	信号音低电平持续时间，单位为毫秒，必须为 8 的整数倍。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

在指定通道上产生单音频或双音频信号音。

注意事项：

- 信号音的频率特征和音量可以通过函数SsmSetTxTonePara设置或配置项DefaultSendToneFrequence、DefaultSendToneVolume进行设置；
- 信号音发生器一旦启动后，线路上将持续出现指定的信号音，直到应用程序调用 SsmStopSendTone。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopSendTone，SsmSetTxTonePara

2.4.2.3 SsmStopSendTone

关闭信号音发生器。

函数原型：

int SsmStopSendTone(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

关闭信号音发生器。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendTone，SsmSendToneEx

2.4.2.4 SsmChkSendTone

查询信号音发生器是否启动，以及正在发送的信号音类型。

函数原型：

int SsmChkSendTone(int ch, int* pnToneType)

参数说明：

ch	通道编号
pnToneType	返回通道上正在发送的信号音类型： 0：拨号音 1：忙音 2：回铃音 3：催挂音 4：自定义信号音； 5：无信号音；

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	通道不在发送信号音
1	通道正在发送信号音

功能描述：

查询信号音发生器是否启动，以及正在发送的信号音类型。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendTone，SsmSendToneEx

2.5 有关信号音检测器的函数

2.5.1 有关信号音检测器的常用函数

2.5.1.1 设置工作状态

2.5.1.1.1 SsmStartToneAnalyze

启动信号音检测器。

函数原型：

int SsmStartToneAnalyze(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

启动信号音检测器。详细内容请参见第 1 章的信号音检测器部分。

注意事项：

- 本函数在默认方式下只适用于模拟外线通道、磁石通道和录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmCloseToneAnalyze，SsmStart2ndToneAnalyzer

2.5.1.1.2 SsmCloseToneAnalyze

关闭信号音分析器。

函数原型：

int SsmCloseToneAnalyze (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

关闭信号音检测器。详细内容请参见第 1 章的信号音检测器部分。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartToneAnalyze

2.5.1.1.3 SsmQueryOpToneAnalyze

查询通道是否具有信号音检测器。

函数原型：

int SsmQueryOpToneAnalyze(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	无
1	有

功能描述：

查询通道是否具有信号音检测器。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.5.1.2 获取检测结果

2.5.1.2.1 SsmGetToneAnalyzeResult

获取信号音检测器的检测结果。

函数原型：

int SsmGetToneAnalyzeResult (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败。
0	信号音分析正在进行。

1	检测到拨号音。
2	检测到忙音。
3	检测到回铃音。
4	检测到回铃音后，线路保持静默。
5	检测到无声。
6	检测到语音应答信号。对于模拟电话线，通常表示被叫用户是否应答（摘机）。
7	检测到具有F1 频率特征的单音频信号音。F1 的参数由函数SsmSetVoiceFxPara设置，通常用于检测传真机的信号音。
8	检测到具有F2 频率特征的单音频信号音。F2 频率参数由函数SsmSetVoiceFxPara设置，通常用于检测传真机的信号音。
9	检测到用户自定义的信号音类型。

功能描述：

获取信号音检测器的检测结果。

注意事项：

- 本函数在默认方式下只适用于模拟外线通道、磁石通道和录音通道；
- 对于数字卡，如要进行信号音分析，需在 ShConfg.ini 文件中的[BoardId＝x]节下对如下配置项进行配置：DefaultToneCheckState=1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmClearToneAnalyzeResult

2.5.1.3 清除检测结果

2.5.1.3.1 SsmClearToneAnalyzeResult

清除信号音检测器的检测结果。

函数原型：

int SsmClearToneAnalyzeResult (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

清除信号音检测器的检测结果。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetToneAnalyzeResult

2.5.2 有关 FFT 的函数

2.5.2.1 设置工作参数

2.5.2.1.1 SsmSetPeakFrqDetectBW

设置检测来话信号中的峰值频率时使用的带宽。

函数原型：

int SsmSetPeakFrqDetectBW(int ch, WORD nPeakBW)

参数说明：

ch	通道编号
nPeakBW	带宽，单位为 Hz，取值范围：40~80，缺省值为 80Hz

返回值：

-1	调用失败
0	调用成功

功能描述：

设置检测来话信号中的峰值频率时使用的带宽。更多信息请参见第 1 章中“信号音检测器”部分内容。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmQueryOpPeakFrqDetect，SsmGetPeakFrqDetectBW

2.5.2.2 获取工作参数

2.5.2.2.1 SsmQueryOpPeakFrqDetect

查询通道是否支持检测峰值频率的操作。

函数原型：

int SsmQueryOpPeakFrqDetect (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	不支持
1	支持

功能描述：

查询通道是否支持检测峰值频率的操作。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.5.2.2.2 SsmGetPeakFrqDetectBW

获取检测峰值频率使用的带宽参数。

函数原型：

int SsmGetPeakFrqDetectBW(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	带宽的设置值

功能描述：

获取检测峰值频率使用的带宽参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmQueryOpPeakFrqDetect，SsmSetPeakFrqDetectBW

2.5.2.3 获取检测结果

2.5.2.3.1 SsmGetPeakFrq

获取来话信号中的峰值频率。

函数原型：

int SsmGetPeakFrqDetectBW(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	来话信号中的峰值频率

功能描述：

获取来话信号中的峰值频率。更多信息请参见第 1 章中“信号音检测器”部分内容。

注意事项：

- E_CHG_PeakFrq 事件可以实现相同的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmQueryOpPeakFrqDetect，SsmSetPeakFrqDetectBW

2.5.2.3.2 SsmGetPeakFrqEnergy

获取来话信号中的峰值频率部分具有的能量值。

函数原型：

long SsmGetPeakFrqEnergy(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	来话信号中的峰值频率部分具有的能量值

功能描述：

获取来话信号中的峰值频率部分具有的能量值。更多信息请参见第 1 章中“信号音检测器”部分内容。

注意事项：

- E_CHG_PeakFrq 事件可以实现相同的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetPeakFrqDetectBW，SsmGetPeakFrq

2.5.2.3.3 SsmGetOverallEnergy

2.5.2.3.4 SsmGetOverallEnergyAllCh

获取来话信号经过 FFT 变换后的全频能量值。SsmGetOverallEnergy 对单个通道进行操作，SsmGetOverallEnergyAllCh 可以同时获取多个通道的全频能量值。

函数原型：

int SsmGetOverallEnergy(int ch)

int SsmGetOverallEnergyAllCh (int nBeginCh, int nChNum, PDWORD pdwEnergyTable)

参数说明：

ch	通道编号
nBeginCh	起始通道编号
nChNum	通道的数量
PdwEnergyTable	存放从 nBeginCh 开始的 nChNum 个通道的能量值，由应用程序分配空间

返回值：

函数名称	返回值	描述
SsmGetOverallEnergy	-1	调用失败
	≥0	全频能量值
SsmGetOverallEnergyAllCh	-1	调用失败
	0	调用成功

功能描述：

获取来话信号经过 FFT 变换后的全频能量值。

注意事项：

- 有关全频能量的单位与DB的换算方法请参见第 1 章中“信号音检测器”节中的“FFT模块”部分。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPeakFrqEnergy

2.5.3 设置噪声滤波器的参数

2.5.3.1 SsmSetMinVocDtrEnergy

设置线路上的噪音判别门限。

函数原型：

int SsmSetMinVocDtrEnergy(int ch, DWORD dwMinVocDtrEnergy)

参数说明：

ch	通道编号
dwMinVocDtrEnergy	噪声判定门限，取值范围：≥700，缺省值为 100000。本参数的单位与DB的换算方法请参见第 1 章中“信号音检测器”节中的“FFT模块”部分。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置线路上的噪音判别门限。如果线路上来话信号的能量值小于 dwMinVocDtrEnergy 的设定值，驱动程序将其视为噪声。

注意事项：

- 配置项MinimumVoiceDetermineEnergy可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMinVocDtrEnergy

2.5.3.2 SsmGetMinVocDtrEnergy

获取驱动程序判定线路上出现的信号是声音还是噪声的能量门限值。

函数原型：

int SsmGetMinVocDtrEnergy(int ch, PDWORD pdwMinVocDtrEnergy)

参数说明：

ch	通道编号
pdwMinVocDtrEnergy	返回判定声音还是噪声的能量门限值

返回值：

-1	调用失败
0	调用成功

功能描述：

获取驱动程序判定线路上出现的信号是声音还是噪声的能量门限值。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetMinVocDtrEnergy

2.5.4 有关呼叫进程音检测器的函数

2.5.4.1 设置第 2 呼叫进程音检测器的工作状态

2.5.4.1.1 SsmStart2ndToneAnalyzer

开启第 2 呼叫进程音检测器。

函数原型：

int SsmStart2ndToneAnalyzer(int ch，BOOL bEnable)

参数说明：

ch	通道编号
bEnable	第 2 组信号音分析器使能开关。=TRUE：开启；=FALSE：关闭

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

开启第 2 呼叫进程音检测器。

注意事项：

- 本函数设置的功能也可以通过配置项Enable2ndToneAnalyzer进行设置；
- 因为第 2 呼叫进程音检测器占用函数SsmSetVoiceFxPara使用的FFT分析器，因此当开启第 2 呼叫进程音检测器后，驱动程序不再检测SsmSetVoiceFxPara函数指定的特定信号音；

- 必须在开启第 1 呼叫进程音检测器的前提下，开启第 2 呼叫进程音检测器才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGet2ndToneAnalyzerState

2.5.4.1.2 SsmGet2ndToneAnalyzerState

查询第 2 呼叫进程音检测器的工作状态。

函数原型：

int SsmGet2ndToneAnalyzerState(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	关闭
1	开启

功能描述：

查询第 2 呼叫进程音检测器的工作状态。

注意事项：

- 本函数在默认方式下只适用于模拟外线通道、磁石通道和录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStart2ndToneAnalyzer

2.5.4.2 获取第 2 呼叫进程音检测器的结果

2.5.4.2.1 SsmGet2ndToneAnalyzeResult

获取第 2 呼叫进程音检测器的结果。

函数原型：

int SsmGet2ndToneAnalyzeResult(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	检测进行中

1	拨号音
2	忙音
3	回铃音
4	回铃音后出现静默
5	无声

功能描述：

获取第 2 呼叫进程音检测器的结果。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmClear2ndToneAnalyzeResult

2.5.4.2.2 SsmClear2ndToneAnalyzeResult

清除第 2 呼叫进程音检测器的结果。

函数原型：

int SsmClear2ndToneAnalyzeResult(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

清除第 2 呼叫进程音检测器的结果。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGet2ndToneAnalyzeResult

2.5.4.3 设置频率检测器的参数

2.5.4.3.1 SsmSetTonePara

2.5.4.3.2 SsmSet2ndTonePara

设置呼叫进程音检测器中频率检测器的工作参数。SsmSetTonePara、SsmSet2ndTonePara 分别用于设置第 1、

第 2 呼叫进程音检测器的频率参数。

函数原型：

int SsmSetTonePara(int ch, WORD wF1, WORD wBw1, WORD wF2, WORD wBw2, DWORD dwMinEnrgRatio)
int SsmSet2ndTonePara(int ch, WORD wF1, WORD wBw1, WORD wF2, WORD wBw2, DWORD dwMinEnrgRatio)

参数说明：

ch	通道编号
wF1	第 1 个中心频率，单位为 Hz，取值范围：300~3400。
wBw1	第 1 个中心频率的带宽，单位为 Hz，取值范围：40~120。
wF2	第 2 个中心频率，单位为 Hz，取值范围：300~3400。
wBw2	第 2 个中心频率的带宽，单位为 Hz，取值范围：40~120。
dwMinEnrgRatio	判定门限，单位：%，取值范围：15~80。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置呼叫进程音检测器中频率检测器的工作参数。详细内容请参见第 1 章中呼叫进程音检测器部分。

注意事项：

- 配置项TonePara可以实现与SsmSetTonePara相同的功能，缺省值为 450Hz单音频信号音；
- 配置项2ndTonePara可以实现与SsmSet2ndTonePara相同的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetTonePara, SsmGet2ndTonePara

2.5.4.3.3 SsmGetTonePara

2.5.4.3.4 SsmGet2ndTonePara

获取呼叫进程音检测器中频率检测器的工作参数。SsmGetTonePara、SsmGet2ndTonePara 分别用于获取第 1、第 2 呼叫进程音检测器的频率参数。

函数原型：

int SsmGetTonePara(int ch, PWORD pwF1, PWORD pwBw1, PWORD pwF2, PWORD pwBw2, PDWORD pdwMinEnrgRatio)
int SsmGet2ndTonePara(int ch, PWORD pwF1, PWORD pwBw1, PWORD pwF2, PWORD pwBw2, PDWORD pdwMinEnrgRatio)

参数说明：

ch	通道编号
pwF1	返回第 1 音频的中心频率，单位为 Hz
pwBw1	返回第 1 音频的带宽，单位为 Hz

pwF2	返回第 2 音频的中心频率，单位为 Hz
pwBw2	返回第 1 音频的带宽，单位为 Hz
pdwMinEnrgRatio	返回设置的判决是否信号音的带内能量占全频能量的百分比门限。

返回值：

-1	调用失败
0	调用成功

功能描述：

获取呼叫进程音检测器中频率检测器的工作参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetTonePara, SsmSet2ndTonePara

2.5.4.4 设置拨号音检测器的参数

2.5.4.4.1 SsmSetIsDialToneDtrTime

2.5.4.4.2 SsmSet2ndIsDialToneDtrTime

设置拨号音检测器的最小保持时间。SsmSetIsDialToneDtrTime、SsmSet2ndIsDialToneDtrTime 分别用于设置第 1、第 2 呼叫进程音检测器。

函数原型：

int SsmSetIsDialToneDtrTime(int ch, WORD wIsDialToneDtrTime)

int SsmSet2ndIsDialToneDtrTime(int ch, WORD wIsDialToneDtrTime)

参数说明：

ch	通道编号
wIsDialToneDtrTime	拨号音的最小保持时间，单位为 ms，取值范围：≥1200ms

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置拨号音检测器的最小保持时间。

注意事项：

- 配置项IsDialToneDetermineTime可以实现与SsmSetIsDialToneDtrTime相同的功能；
- 配置项2ndIsDialToneDetermineTime可以实现与SsmSetIsDialToneDtrTime相同的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：SsmGetIsDialToneDtrTime, SsmGet2ndIsDialToneDtrTime

2.5.4.4.3 SsmGetIsDialToneDtrTime

2.5.4.4.4 SsmGet2ndIsDialToneDtrTime

获取拨号音检测器的最小保持时间。SsmGetIsDialToneDtrTime、SsmGet2ndIsDialToneDtrTime分别用于设置第 1、第 2 呼叫进程音检测器。

函数原型：

int SsmGetIsDialToneDtrTime(int ch, PWORD pwIsDialToneDtrTime)

int SsmGet2ndIsDialToneDtrTime(int ch, PWORD pwIsDialToneDtrTime)

参数说明：

ch	通道编号
pwIsDialToneDtrTime	返回最小保持时间，单位为毫秒

返回值：

-1	调用失败
0	调用成功

功能描述：

获取拨号音检测器的最小保持时间。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIsDialToneDtrTime，SsmSet2ndIsDialToneDtrTime

2.5.4.5 设置回铃音检测器的参数

2.5.4.5.1 SsmSetRingEchoTonePara

2.5.4.5.2 SsmSet2ndRingEchoTonePara

设置回铃音检测器的高电平保持时间和低电平保持时间。SsmSetRingEchoTonePara 适用于第 1 呼叫进程音检测器，SsmSet2ndRingEchoTonePara 适用于第 2 呼叫进程音检测器。

函数原型：

int SsmSetRingEchoTonePara(int ch, WORD wTon, WORD wToff)

int SsmSet2ndRingEchoTonePara(int ch, WORD wTon, WORD wToff)

参数说明：

ch	通道编号
wTon	高电平保持时间，单位为毫秒，取值范围： 500~2500。
wToff	低电平保持时间，单位为毫秒，取值范围： 1500 ~ 6000。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置回铃音检测器的高电平保持时间和低电平保持时间。

注意事项：

- SsmSetRingEchoTonePara设置的参数可以通过配置项RingEchoTonePara进行设置，缺省值：高电平保持时间为 1000 毫秒，低电平保持时间为 4000 毫秒；SsmSet2ndRingEchoTonePara设置的参数可以通过配置项2ndRingEchoTonePara进行设置，缺省值：高电平保持时间为 1000 毫秒，低电平保持时间为 4000 毫秒。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRingEchoTonePara，SsmGet2ndRingEchoTonePara

2.5.4.5.3 SsmGetRingEchoToneTime

获取回铃音的保持时间。

函数原型：

int SsmGetRingEchoToneTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	保持时间，单位：毫秒

功能描述：

获取回铃音的保持时间。

注意事项：

- 本函数只适用于模拟中继线通道

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.5.4.5.4 SsmGetRingEchoTonePara

2.5.4.5.5 SsmGet2ndRingEchoTonePara

获取回铃音检测器的高电平保持时间和低电平保持时间。SsmGetRingEchoTonePara、SsmGet2ndRingEchoTonePara 分别用于第 1、第 2 呼叫进程音检测器。

函数原型：

int SsmGetRingEchoTonePara(int ch, PWORD pwTon, PWORD pwToff)

int SsmGet2ndRingEchoTonePara(int ch, PWORD pwTon, PWORD pwToff)

参数说明：

ch	通道编号
pwTon	返回高电平保持时间，单位为毫秒
pwToff	返回低电平保持时间，单位为毫秒

返回值：

-1	调用失败
0	调用成功

功能描述：

获取回铃音检测器的高电平保持时间和低电平保持时间。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRingEchoTonePara，SsmSet2ndRingEchoTonePara

2.5.4.6 设置忙音检测器的参数

2.5.4.6.1 SsmSetBusyTonePeriodEx

2.5.4.6.2 SsmSetBusyTonePeriod

2.5.4.6.3 SsmSet2ndBusyTonePeriod

设置呼叫进程音检测器中忙音检测器的忙音周期。SsmSetBusyTonePeriod 适用于第 1 呼叫进程音检测器，SsmSet2ndBusyTonePeriod 适用于第 2 呼叫进程音检测器，SsmSetBusyTonePeriodEx 函数可以同时适用于第 1、第 2 呼叫进程音检测器。

函数原型：

int SsmSetBusyTonePeriodEx(int ch, int nCptdNo, WORD wMax, PWORD pwPeriod)

int SsmSetBusyTonePeriod(int ch, WORD wPeriod)

int SsmSet2ndBusyTonePeriod(int ch, WORD wPeriod)

参数说明：

ch	通道编号
nCptdNo	=1：设置第 1 组呼叫进程音检测器 =2：设置第 2 组呼叫进程音检测器
wMax	忙音周期的个数。取值范围：1~4
pwPeriod	存放忙音周期参数的指针，由应用程序提供存储空间
wPeriod	忙音周期，单位为毫秒，取值范围：200~2000。本参数也可在配置文件中指定，缺省值为 700ms (350ms/ON，350ms/OFF)

返回值：

-1	调用失败
0	调用成功

功能描述：

设置呼叫进程音检测器中忙音检测器的忙音周期。SsmSetBusyTonePeriodEx是新版本的函数，功能更加强大，可以设置多达四个不同的忙音周期。更多信息请参见第 1 章中“忙音检测器”部分内容。

注意事项：

- 忙音周期也可以分别通过配置项BusyTonePeriod或2ndBusyTonePeriod 进行设置；
- 建议使用 SsmSetBusyTonePeriodEx 函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetBusyTonePeriodEx，SsmGetBusyTonePeriod，SsmGet2ndBusyTonePeriod

代码范例：

系统需要在第 1 呼叫进程音检测器上同时检测周期分别为 700 毫秒、800 毫秒、900 毫秒、1000 毫秒的 4 种忙音，在第 2 呼叫进程音检测器上只检测周期为 700 毫秒的忙音，实现代码为：

```
WORD BusyTonePeriod[4] = {700,800,900,1000};
SsmSetBusyTonePeriodEx(ch, 1, 4, BusyTonePeriod);
SsmSet2ndBusyTonePeriod(ch, 700);
```

2.5.4.6.4 SsmSetIsBusyToneDtrCnt

2.5.4.6.5 SsmSet2ndIsBusyToneDtrCnt

设置忙音周期的最少持续个数。SsmSetIsBusyToneDtrCnt 适用于第 1 呼叫进程音检测器，SsmSet2ndIsBusyToneDtrCnt 适用于第 2 呼叫进程音检测器。

函数原型：

int SsmSetIsBusyToneDtrCnt(int ch, WORD wIsBusyToneDtrCnt)

int SsmSet2ndIsBusyToneDtrCnt(int ch, WORD wIsBusyToneDtrCnt)

参数说明：

ch	通道编号
wIsBusyToneDtrCnt	忙音周期的最小持续个数，取值范围：1~50

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置忙音周期的最少持续个数。更多信息请参见第 1 章中忙音检测器部分。

注意事项：

- 配置项IsBusyToneDetermineCount、 2ndIsBusyToneDetermineCount也可实现同样功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIsBusyToneDtrCnt，SsmGet2ndIsBusyToneDtrCnt

2.5.4.6.6 SsmGetBusyTonePeriodEx

2.5.4.6.7 SsmGetBusyTonePeriod

2.5.4.6.8 SsmGet2ndBusyTonePeriod

获取忙音判定周期的设定值。SsmGetBusyTonePeriod 适用于第 1 呼叫进程音检测器，SsmGet2ndBusyTonePeriod 适用于第 2 呼叫进程音检测器，SsmGetBusyTonePeriodEx 函数可以同时适用于第 1、第 2 呼叫进程音检测器。

函数原型：

int SsmGetBusyTonePeriodEx(int ch, int nCptdNo, PWORD pwPeriod)

int SsmGetBusyTonePeriod(int ch, PWORD pwBusyTonePeriod)

int SsmGet2ndBusyTonePeriod(int ch, PWORD pwBusyTonePeriod)

参数说明：

ch	通道编号
nCptdNo	=1：设置第 1 组呼叫进程音检测器 =2：设置第 2 组呼叫进程音检测器
pwPeriod	返回忙音周期设定值的指针，单位为毫秒，由应用程序提供存储空间，大小为 4 WORD
pwBusyTonePeriod	返回忙音周期设定值，单位为毫秒

返回值：

-1	调用失败
0	调用成功
>0	设置的忙音周期个数

功能描述：

获取忙音判定周期的设定值。更多信息请参见第 1 章中“忙音检测器”部分内容。

注意事项：

- 建议使用 SsmGetBusyTonePeriodEx 函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetBusyTonePeriodEx，SsmSetBusyTonePeriod，SsmSet2ndBusyTonePeriod

2.5.4.6.9 SsmGetIsBusyToneDtrCnt

2.5.4.6.10 SsmGet2ndIsBusyToneDtrCnt

获取忙音周期最少持续个数的设定值。SsmGetIsBusyToneDtrCnt 适用于第 1 呼叫进程音检测器，SsmGet2ndIsBusyToneDtrCnt 适用于第 2 呼叫进程音检测器。

函数原型：

int SsmGetIsBusyToneDtrCnt(int ch, PWORD pwIsBusyToneDtrCnt)

int SsmGet2ndIsBusyToneDtrCnt(int ch, PWORD pwIsBusyToneDtrCnt)

参数说明：

ch	通道编号
pwIsBusyToneDtrCnt	返回设置的忙音周期的最少持续个数

返回值：

-1	调用失败
0	调用成功

功能描述：

获取忙音周期最少持续个数的设定值。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIsBusyToneDtrCnt，SsmSetIsBusyToneDtrCnt

2.5.4.7 获取检测结果

2.5.4.7.1 SsmGetBusyToneLen

2.5.4.7.2 SsmGet2ndBusyToneLen

获取忙音的保持时间。SsmGetBusyToneLen 适用于第 1 呼叫进程音检测器，SsmGet2ndBusyToneLen 适用于第 2 呼叫进程音检测器。

函数原型：

int SsmGetBusyToneLen(int ch)

int SsmGet2ndBusyToneLen(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	忙音持续的时间，单位为秒

功能描述：

获取忙音的保持时间。更多信息请参见第 1 章中“忙音检测器”部分内容。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetBusyToneCount

2.5.4.7.3 SsmGetBusyToneCount

2.5.4.7.4 SsmGet2ndBusyToneCount

获取忙音周期的个数。SsmGetBusyToneCount 适用于第 1 呼叫进程音检测器，SsmGet2ndBusyToneCount 适用于第 2 呼叫进程音检测器。

函数原型：

int SsmGetBusyToneCount(int ch)

int SsmGet2ndBusyToneCount(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	忙音周期持续的个数

功能描述：

获取忙音周期的个数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetBusyToneLen

2.5.4.8 获取采用背靠背忙音信号检测算法的检测结果

2.5.4.8.1 SsmGetBusyToneEx

获取忙音检测器采用背靠背忙音信号检测算法时的检测结果。

函数原型：

int SsmGetBusyToneEx(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	没有检测到忙音信号
1	检测到忙音信号

功能描述：

获取忙音检测器采用背靠背忙音信号检测算法时的检测结果。更多信息请参见第 1 章中“忙音检测器”部分内容。

注意事项：

- 使用 E_CHG_BusyToneEx 事件可以实现同样的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmClearToneAnalyzeResult

2.5.5 有关传真进程音检测器的函数

2.5.5.1 设置工作参数

2.5.5.1.1 SsmSetVoiceFxPara

设置传真进程音检测器的工作参数。

函数原型：

int SsmSetVoiceFxPara(int ch, WORD wSelFx, WORD wFx,WORD wFxBW, DWORD dwIsVocFxRatio, WORD wIsVocFxDtrTime)

参数说明：

ch	通道编号
wSelFx	选择传真进程音检测器，取值范围： 1：设置第 1 传真进程音检测器的参数 2：设置第 2 传真进程音检测器的参数
wFx	中心频率，单位：Hz，取值范围：300~3400
wFxBW	带宽，单位：Hz，取值范围：40~120
dwIsVocFxRatio	带内能量占全频能量的百分比的最小门限值，单位：%，取值范围：15 ~ 80
wIsVocFxDtrTime	最小保持时间，单位：毫秒，取值范围：≥64

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置传真进程音检测器的工作参数。更多信息请参见第 1 章中“传真进程音检测器”部分内容。

注意事项：

- 配置项VoiceFreqF1Para、VoiceFreqF2Para可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetVocFxFlag，SsmGetVoiceFxPara

2.5.5.1.2 SsmGetVoiceFxPara

获取传真进程音检测器的工作参数。

函数原型：

int SsmGetVoiceFxPara(int ch, WORD wSelFx, PWORD pwFx, PWORD pwFxBW, PDWORD pdwIsVocFxRatio, PWORD pwIsVocFxDtrTime)

参数说明：

ch	通道编号
wSelFx	选择传真进程音检测器，取值范围： 1：设置第 1 传真进程音检测器的参数 2：设置第 2 传真进程音检测器的参数
pwFx	返回中心频率，单位：Hz
pwFxBW	返回带宽，单位：Hz
pdwIsVocFxRatio	返回带内能量占全频能量的百分比的最小门限值，单位：%
wIsVocFxDtrTime	返回最小保持时间，单位：毫秒

返回值：

-1	调用失败
0	调用成功

功能描述：

获取传真进程音检测器的工作参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetVoiceFxPara

2.5.5.2 获取检测结果

2.5.5.2.1 SsmGetVocFxFlag

获取传真进程音检测器的检测结果。

函数原型：

int SsmGetVocFxFlag(int ch, int nSelFx, BOOL bClear)

参数说明：

ch	通道编号
nSelFx	选择检测的声音频率。 =1：选择 F1；=2：选择 F2
bClear	指示驱动程序在读取指定标志后是否清除检测结果。 =TRUE：清除检测结果；=FALSE:：不清除检测结果。

返回值：

-1	调用失败
0	没有发现指定频率的单音频声音
1	发现指定频率的单音频声音

功能描述：

获取传真进程音检测器的检测结果。更多信息请参见第 1 章中“传真进程音检测器”部分内容。

注意事项：

- 建议使用事件 E_CHG_VocFxFlag 来实现同样功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetVoiceFxPara

2.6 有关 DTMF 检测器的函数

驱动程序内部有接收 DTMF 字符串缓冲区，缓冲区大小可以在配置文件中指定，缺省值为 200。应用程序应当在驱动程序的缓冲区溢出前及时读走 DTMF 字符，否则，如果驱动程序再次收到一个新的字符，将会导致最早收到的 DTMF 字符丢失。

对于坐席通道和录音通道，驱动程序检测到摘机或挂机时将自动清空接收 DTMF 缓冲区； 对于其它类型通道，驱动程序在收到挂机命令时自动清空接收缓冲区。

本节所有函数不支持磁石通道。

2.6.1 启动/关闭 DTMF 检测器

2.6.1.1 SsmEnableRxDtmf

启动或关闭 DTMF 检测器。

函数原型：

int SsmEnableRxDtmf(int ch, BOOL bRun)

参数说明：

ch	通道编号
bRun	是否开启 DTMF 检测器的控制开关。 TRUE： 开启 FALSE：关闭

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

启动或关闭 DTMF 检测器。

注意事项：

- 驱动程序会根据通道的状态迁移自动调用本函数。只有当应用程序自行控制呼叫过程时才需要调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRxDtmfHandler，SsmGetDtmfStr

2.6.2 设置回调函数

2.6.2.1 SsmSetRxDtmfHandler

设置DTMF检测器输出DTMF信号时的回调函数。

函数原型：

int SsmSetRxDtmfHandler(int ch, RXDTMFHANDLER pfnOnRcvDtmf, PVOID pV)

参数说明：

ch	通道编号
pfnOnRcvDtmf	回调函数的指针。回调函数类型为 RXDTMFHANDLER，在 shpa3api.h 中声明为： typedef void (*RXDTMFHANDLER) (int **ch**, char **cDtmf**, int **nDTStatus**, PVOID **pV**); 上式中， **ch**： 通道编号； **cDtmf**： DTMF 字符（ASCII 码形式）； **nDTStatus**：DTMF 信号类型，0 表示 DTMF 信号消失，1 表示 DTMF 信号出； **pV**： PVOID 类型指针，通常用于应用程序向回调函数传递其数据结构。这个指针将由驱动程序透明地传递给回调函数。
pV	PVOID 类型指针。应用程序可以通过本参数向回调函数传递需要的对象或数据结构

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

设置DTMF检测器输出DTMF信号时的回调函数。

注意事项：

- DTMF检测器每检测一个完整的DTMF字符，会调用 2 次本函数设置的回调函数，一次是DTMF信号出现的时候，另一次是DTMF信号消失的时候。更多信息请参见第 1 章中的DTMF检测器部分；
- 收到的 DTMF 字符仍然会进入 DTMF 检测器的缓冲区。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.6.3 获取 DTMF 字符

2.6.3.1 SsmClearRxDtmfBuf

清空驱动程序内部的 DTMF 按键号码接收缓冲区。

函数原型：

int SsmClearRxDtmfBuf(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

清空驱动程序内部的 DTMF 接收缓冲区。

注意事项：

- 当通道迁移到空闲状态时，驱动程序会自动清空 DTMF 检测器的缓冲区。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetDtmfStr

2.6.3.2 SsmGetDtmfStr

2.6.3.3 SsmGetDtmfStrA

获取保存在 DTMF 检测器缓冲区的 DTMF 字符。

函数原型：

int SsmGetDtmfStr(int ch, LPSTR pszDtmf)

char* SsmGetDtmfStrA(int ch)

参数说明：

ch	通道编号
pszDtmf	存放接收 DTMF 字符串的指针，由应用程序分配存储空间

返回值：

函数名称	返回值	描述
SsmGetDtmfStr	-1	调用失败
	≥0	DTMF 检测器中的字符个数
SsmGetDtmfStrA	NULL	调用失败

	其它	返回 DTMF 检测器缓冲区的指针

功能描述：

获取保存在DTMF检测器缓冲区的DTMF字符，字符为标准的ASCII码。SsmGetDtmfStr和SsmGetDtmfStrA功能完全一样，只是返回DTMF检测器缓冲区的方式不一样。

注意事项：

- 本函数调用时，驱动程序不会清空 DTMF 检测器缓冲区。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRxDtmfLen，SsmGet1stDtmf，SsmGet1stDtmfClr，SsmGetLastDtmf

2.6.3.4 SsmGetRxDtmfLen

获取 DTMF 检测器缓冲区中保存的 DTMF 字符个数。

函数原型：

int SsmGetRxDtmfLen(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
≥0	DTMF 检测器缓冲区中保存的 DTMF 字符个数

功能描述：

获取 DTMF 检测器缓冲区中保存的 DTMF 字符个数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetDtmfStr，SsmGetDtmfStrA

2.6.3.5 SsmGet1stDtmf

2.6.3.6 SsmGet1stDtmfClr

获取 DTMF 检测器缓冲区中最先收到的 DTMF 字符。

函数原型：

int SsmGet1stDtmf(int ch, char* pcDtmf)

int SsmGet1stDtmfClr(int ch, char* pcDtmf)

参数说明：

ch	通道编号
pcDtmf	存放 DTMF 字符的缓冲区首地址

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	DTMF 检测器缓冲区为空
1	DTMF 检测器缓冲区中有字符

功能描述：

获取 DTMF 检测器缓冲区中最先收到的 DTMF 字符。SsmGet1stDtmf 在获取 DTMF 字符后，该字符仍然保留在缓冲区中；SsmGet1stDtmfClr 则会将取到的字符从缓冲区中清除。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetDtmfStr，SsmGetDtmfStrA

2.6.3.7 SsmGetLastDtmf

获取 DTMF 检测器缓冲区中最后收到的 DTMF 字符。

函数原型：

int SsmGetLastDtmf(int ch, char* pcDtmf)

参数说明：

ch	通道编号
pcDtmf	存放 DTMF 字符的缓冲区首地址

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	DTMF 检测器缓冲区为空
1	DTMF 检测器缓冲区中有字符

功能描述：

获取 DTMF 检测器缓冲区中最后收到的 DTMF 字符。

注意事项： 无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRxDtmfLen，SsmGet1stDtmfClr，SsmGetLastDtmf，SsmGetDtmfStr，SsmGetDtmfStrA

2.6.4 使用 WaitDtmf 任务

2.6.4.1 SsmSetWaitDtmf

2.6.4.2 SsmSetWaitDtmfEx

2.6.4.3 SsmSetWaitDtmfExA

启动 WaitDtmf 任务。SsmSetWaitDtmf 的最大等待时间的精度为秒级，而 SsmSetWaitDtmfEx 和 SsmSetWaitDtmfExA 的最大等待时间的精度为毫秒级；SsmSetWaitDtmfEx 只支持 1 个终止字符，而 SsmSetWaitDtmfExA 支持多个终止字符。

函数原型：

int SsmSetWaitDtmf(int ch, WORD wTimeOut1, WORD wMaxLen, char cEndChar, BOOL bWithEndChar)

int SsmSetWaitDtmfEx(int ch, WORD wTimeOut2, WORD wMaxLen, char cEndChar, BOOL bWithEndChar)

int SsmSetWaitDtmfExA(int ch, WORD wTimeOut2, WORD wMaxLen, char* szEndChar, BOOL bWithEndChar)

参数说明：

ch	通道编号
wTimeOut1	最大等待时间，单位：秒。
wTimeOut2	最大等待时间，单位：毫秒
wMaxLen	等待接收 DTMF 的最大长度，取值范围：1~50
cEndChar	结束字符。如果驱动程序收到此字符，WaitDtmf 任务结束。
szEndChar	结束字符集，最多可设值 16 个。如果驱动程序收到此字符集中的任何一个字符，WaitDtmf 任务结束。
bWithEndChar	WaitDtmf 任务结束后，驱动程序返回给应用程序的字符串中是否包含收到的结束字符由本参数决定： =TRUE： 包括 =FALSE： 不包括

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	设置成功

功能描述：

向驱动程序提交一个接收特定 DTMF 字符串的任务，即 WaitDtmf 任务。任务成功启动后，驱动程序对 DTMF 接收缓冲区进行检查，直到下列条件满足才注销该任务：

- ✧ 收到由参数 cEndChar 或 szEndChar 中指定的结束字符；
- ✧ 已经接收的 DTMF 数量等于参数 wMaxLen 设定的最大接收长度；
- ✧ 等待时间超出参数 wTimeOut1 或 wTimeOut2 所指定的时间。

当WaitDtmf任务结束后，驱动程序会向应用程序抛出E_PROC_WaitDTMF事件。任务的执行结果可以通过函数SsmChkWaitDtmf得到。

函数SsmCancelWaitDtmf可以用来取消WaitDtmf任务。

注意事项：

- 驱动程序在指定通道上每当收到一个 DTMF 字符，会自动清零定时器，重新开始计时；
- 调用本函数时，不会清除 DTMF 缓冲区中以前收到的字符。即缓冲区中残留的字符会影响本函数执行结果；
- 如果通道上启动了 WaitDtmf 任务，收到的 DTMF 字符将不再进入 DTMF 检测器的接收缓冲区。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmChkWaitDtmf，SsmCancelWaitDtmf

2.6.4.4 SsmCancelWaitDtmf

取消 WaitDtmf 任务。

函数原型：

int SsmCancelWaitDtmf(int ch)

参数说明：

ch	通道编号

返回值：

-1	出现错误
0	调用成功

功能描述：

取消 WaitDtmf 任务。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetWaitDtmf，SsmSetWaitDtmfEx，SsmSetWaitDtmfExA，SsmChkWaitDtmf

2.6.4.5 SsmChkWaitDtmf

查询 WaitDtmf 任务的进展情况。

函数原型：

int SsmChkWaitDtmf(int ch, LPSTR pszDtmf)

参数说明：

ch	通道编号
pszDtmf	存放 DTMF 的缓冲区指针，由应用程序分配存储空间。

返回值：

-1	出现错误
0	正在接收
1	接收超时，任务结束
2	任务由于接收到指定的结束字符而结束
3	任务由于接收到指定长度的 DTMF 字符串而结束

功能描述：

查询 WaitDtmf 任务的进展情况。

注意事项：

- E_PROC_WaitDTMF 事件提供相同的功能

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：E_PROC_WaitDTMF

2.7 有关 DTMF 发生器的函数（CTI 系列）

2.7.1 设置 DTMF 发生器的参数

2.7.1.1 SsmSetTxDtmfPara

设置 DTMF 发生器产生的 DTMF 字符的高电平保持时间和低电平保持时间。

函数原型：

int SsmSetTxDtmfPara(int ch, WORD wOnTime, WORD wOffTime)

参数说明：

ch	通道编号
wOnTime	高电平保持时间，单位为毫秒，取值范围：≥40，必须为 8 的整数倍。
wOffTime	低电平保持时间，单位为毫秒，取值范围：≥40，必须为 8 的整数倍。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置DTMF发生器产生的DTMF字符的高电平保持时间和低电平保持时间。

注意事项：

- 配置项TxDtmfTimePara可以实现相同功能。
- 对 IP 通道而言，仅在采用带内方式发送时调用本函数才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetTxDtmfPara，SsmTxDtmf

2.7.1.2 SsmQueryTxDtmf

查询通道是否具有DTMF发生器。

函数原型：

int SsmQueryTxDtmf(int ch, LPSTR pszReserved)

参数说明：

ch	通道编号
PszReserved	保留的字符串指针，可不使用

返回值：

-1	调用失败
0	指定通道不支持发送标准 DTMF 字符的操作
1	指定通道支持发送标准 DTMF 字符的操作

功能描述：

查询通道是否具有DTMF发生器。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.7.1.3 SsmGetTxDtmfPara

获取 DTMF 发生器产生的 DTMF 字符的高电平保持时间和低电平保持时间。

函数原型：

int SsmGetTxDtmfPara(int ch, PWORD pwOnTime, PWORD pwOffTime)

参数说明：

ch	通道编号
pwOnTime	返回高电平保持时间，单位为毫秒
pwOffTime	返回低电平保持时间，单位为毫秒

返回值：

-1	调用失败
=0	调用成功

功能描述：

获取 DTMF 发生器产生的 DTMF 字符的高电平保持时间和低电平保持时间。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetTxDtmfPara

2.7.2 启动 DTMF 发生任务

2.7.2.1 SsmTxDtmf

在线路的去话方向上发送 DTMF 字符串。

函数原型：

int SsmTxDtmf(int ch, LPSTR pszDtmf)

参数说明：

ch	通道编号
pszDtmf	● 字符串的指针，必须以'0'字符结束，字符串的个数不能超过配置项TxDtmfBufSize的设定值。字符的取值范围： a) '0'~'9','*','#','A','B','C','D'：标准的 DTMF 信号； b) '!'：在线路上产生一次 flash，只适用于模拟中继线通道； c) 其它字符：在线路上保持一段时间的静默。保持时间由配置项DefaultTxDelayTime设置，缺省值为 2000 毫秒。 ● 对于 IP 通道，本参数的意义随 DTMF 发送方式不同而有以下变化： a) 采用带内、RFC2833 事件发送方式时，仅支持'0'~'9','*','#','A','B','C','D'这 15 个标准的 DTMF 字符；

	b) 采用带外发送方式时，可使用 ASCII 字符集中的任意字符。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

在线路的去话方向上发送 DTMF 字符串。

驱动程序将pszDtmf中的字符拷贝到DTMF发生器内部的发送缓冲区后，立即启动DTMF发送任务。当发送缓冲区中的全部字符发送完毕后，驱动程序会向应用程序抛出E_PROC_SendDTMF事件。函数SsmChkTxDtmf也可以用来查询发送任务的进展情况，函数SsmStopTxDtmf可以用来终止发送任务。

注意事项：

- DTMF发生器的发送缓冲区大小可以通过配置项TxDtmfBufSize设置，缺省值为 50 个字符；
- 本函数只支持 SHT 系列、SHD 系列、SHN 系列板卡。
- 对 IP 通道而言，本函数需注意以下几点：
  a) 采用带外方式发送时，本函数是同步函数。采用其他方式发送时是异步函数。
  b) 只有在采用带内方式发送时，调用函数SsmStopTxDtmf结束发送过程才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmChkTxDtmf，SsmStopTxDtmf，SsmSetTxDtmfPara

2.7.2.2 SsmStopTxDtmf

取消 DTMF 发生器的发送任务。

函数原型：

int SsmStopTxDtmf(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

取消 DTMF 发生器的发送任务。

注意事项：

- 对 IP 通道而言，仅在采用带内方式发送时调用本函数才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmTxDtmf

2.7.2.3 SsmChkTxDtmf

查询 DTMF 发生器发送任务的进展情况。

函数原型：

int SsmChkTxDtmf(int ch)

参数说明：

ch	通道编号

返回值：

-1	出现错误
0	发送 DTMF 字符串的任务完成
1	发送 DTMF 字符串的正在进行

功能描述：

查询 DTMF 发生器发送任务的进展情况。

注意事项：

- E_PROC_SendDTMF 事件可以实现相同功能。
- 对 IP 通道而言，仅在采用带内方式发送时调用本函数才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmTxDtmf，SsmStopTxDtmf

2.8 有关 Barge-in 检测器的函数

2.8.1 设置 Barge-in 检测器的参数

2.8.1.1 SsmSetBargeInSens

设置 Barge In 检测器的灵敏度。

函数原型：

int SsmSetBargeInSens(int ch, int nSens)

参数说明：

ch	通道编号
nSens	敏感度，取值范围：0~31，值越大越灵敏

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置Barge In检测器的灵敏度，更多信息请参见第 1 章中Barge in检测器部分。

注意事项：

- 配置项BargeInSensitive可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetBargeInSens，SsmSetIsBargeInDtrmTime，SsmSetNoSoundDtrmTime

2.8.1.2 SsmSetIsBargeInDtrmTime

设置 BargeIn 的最小保持时间。

函数原型：

int SsmSetIsBargeInDtrmTime(int ch, WORD wMinKeepTime)

参数说明：

ch	通道编号
wMinKeepTime	最小保持时间，单位：毫秒，取值范围：≥16，必须为 16 毫秒的整数倍

返回值：

-1	不支持或参数设置出错
0	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
1	调用成功

功能描述：

设置 BargeIn 的最小保持时间。

当来话中检测到语音信号，并且保持时间大于本函数的设定值，BargeIn检测器会判决输出“Barge In”的检测结果，并向应用程序抛出E_SYS_BargeIn事件。更多信息请参见第 1 章中“Barge in检测器”部分内容。

注意事项：

- 配置项BargeInDtrmTime可以实现相同功能，缺省值为 32 毫秒。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmGetIsBargeInDtrmTime，SsmSetBargeInSens，SsmSetNoSoundDtrmTime

2.8.1.3 SsmSetNoSoundDtrmTime

设置静默的最小保持时间。

函数原型：

int SsmSetNoSoundDtrmTime(int ch, DWORD wMinKeepTime)

参数说明：

ch	通道编号
wMinKeepTime	最小保持时间，单位：毫秒，取值范围：≥16，必须为 16 毫秒的整数倍

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置静默的最小保持时间。

当来话中的语音信号消失（即保持静默），并且保持时间大于本函数的设定值，BargeIn检测器会判决输出“无声”的检测结果，并向应用程序抛出E_SYS_NoSound事件。更多信息请参见第 1 章中“Barge in检测器”部分内容。

注意事项：

- 配置项IsNoSoundDtrmTime可以实现相同功能，缺省值为 5000 毫秒。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIsBargeInDtrmTime，SsmSetBargeInSens，SsmGetNoSoundDtrmTime

2.8.1.4 SsmGetBargeInSens

获取 Barge In 检测器的灵敏度的设定值。

函数原型：

int SsmGetBargeInSens(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
≥0	Barge In 检测器的灵敏度

功能描述：

获取 Barge In 检测器的灵敏度的设定值。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetBargeInSens

2.8.1.5 SsmGetIsBargeInDtrmTime

获取 BargeIn 的最小保持时间的设定值。

函数原型：

int SsmGetIsBargeInDtrmTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	最小保持时间，单位为毫秒

功能描述：

获取 BargeIn 的最小保持时间的设定值。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIsBargeInDtrmTime

2.8.1.6 SsmGetNoSoundDtrmTime

获取静默的最小保持时间的设定值。

函数原型：

int SsmGetNoSoundDtrmTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	最小保持时间，单位为毫秒

功能描述：

获取静默的最小保持时间的设定值。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetNoSoundDtrmTime

2.8.2 获取 Barge-in 检测器的检测结果

2.8.2.1 SsmDetectBargeIn

获取 Bargein 检测器的检测结果。

函数原型：

int SsmDetectBargeIn(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	没有检测到 Bargein
1	检测到 Bargein

功能描述：

获取Bargein检测器的检测结果。更多信息请参见第 1 章中“Barge in检测器”部分内容。

注意事项：

- 推荐使用 E_SYS_BargeIn 事件来获取检测结果。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmDetectNoSound

2.8.2.2 SsmDetectNoSound

函数原型：

int SsmDetectNoSound(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	没有检测到”无声”
1	检测到”无声”

功能描述：

获取 Bargein 检测器的检测结果。

注意事项：

- 推荐使用 E_SYS_NoSound 事件来获取检测结果。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmDetectBargeIn

2.8.2.3 SsmGetNoSoundTime

获取无声的保持时间。

函数原型：

int SsmGetNoSoundTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	无声的保持时间，单位为毫秒

功能描述：

获取无声的保持时间。更多信息请参见第 1 章中“Barge in检测器”部分内容。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmDetectNoSound

2.9 有关放音操作的函数

2.9.1 设置放音数据的去向

2.9.1.1 SsmSetPlayDest

设置是否将放音数据送到 TDM 交换总线上。

函数原型：

int SsmSetPlayDest(int ch, int nSelDest)

参数说明：

ch	通道编号
nSelDest	放音数据是否送上 TDM 交换总线的选择开关。对于 SHD 系列板卡，本参数控制 SHD 系列板卡的工作原理框图中开关 K1-1 的状态；对于 SHT 系列板卡，本参数控制 SHT 系列板卡的工作原理框图中开关 K3 的状态。取值范围： 0：不送上总线（开关断开）； 1：送上总线（开关闭合）。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置是否将放音数据送到TDM交换总线上。更多信息请参见第 1 章中“SHD系列板卡的工作原理框图”或“SHT系列板卡的工作原理框图”部分内容。

注意事项：

- 本函数只能用于电话会议中给会议室播放背景音的目的；
- 本函数只支持 SHT 系列、SHD 系列、SHN 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.9.2 设置放音的音量

2.9.2.1 SsmSetPlayVolume

2.9.2.2 SsmSetPlayGain

设置 SHD/SHT/SHN 系列板卡的放音音量。SsmSetPlayGain 提供比 SsmSetPlayVolume 更为精细的控制。

函数原型：

int SsmSetPlayVolume(int ch, int nGain)

int SsmSetPlayGain(int ch, WORD wGainLevel)

参数说明：

ch	通道编号
nGain	音量的增益，取值范围：-7~+6。大于 0 表示提升，小于 0 表示衰减，-7 表示完全关闭声音输出，缺省值为 0。音量值乘以 3 即为分贝值。
wGainLevel	16 进制方式表达的音量的增益，低 8 比特有效，高 8 比特保留，应设置为 0。取值范围： wGainLevel=0：　完全关闭声音输出 wGainLevel=1：　缩小 16 倍； wGainLevel=16：　增益保持不变（0DB）； 32≤wGainLevel≤255：　除以 16 后得到的数即为增益放大倍数； wGainLevel=255：　放大 16 倍。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

2 个函数均用来设置SHD/SHT/ATP/SHN系列板卡中音量调节器A1 的增益。有关音量调节器A1 的详细内容请参见SHD系列板卡的工作原理框图或SHT系列板卡的工作原理框图部分。

注意事项：

- 配置项DefaultPlayVolume也可以实现SsmSetPlayVolume的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.9.3 设置播放任务的终止条件

2.9.3.1 SsmSetDtmfStopPlay

设置放音任务是否因DTMF检测器检测到DTMF字符而终止。

函数原型：

int SsmSetDtmfStopPlay(int ch, BOOL bTerminatePlaybackOnDTMF)

参数说明：

ch	通道编号
bTerminatePlaybackOnDTMF	使能标志，取值范围： TRUE：　开启 FALSE：关闭

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置放音任务是否因DTMF检测器检测到DTMF字符而终止。

如果参数bTerminatePlaybackOnDTMF设置为TRUE，当通道上正在执行放音任务时，如果DTMF检测器在来话信号中检测到了DTMF字符，并且该DTMF字符包含在用SsmSetDTMFStopPlayCharSet函数或配置项

DtmfStopPlayCharSet预设的字符集中，驱动程序就会立即终止放音任务。

注意事项：

- 配置项DtmfStopPlayCharSet也可以实现相同功能，缺省值为开启此项功能；
- 本函数设置的标志对预播放加载文件、单个文件、文件序列、单个内存、乒乓内存和内存序列均起作用。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDTMFStopPlayCharSet，SsmGetDtmfStopPlayFlag

2.9.3.2 SsmSetDTMFStopPlayCharSet

设置终止放音任务的 DTMF 字符集。

函数原型：

int SsmSetDTMFStopPlayCharSet(int ch, LPSTR lpstrDtmfCharSet)

参数说明：

ch	通道编号
lpstrDtmfCharSet	终止放音任务的 DTMF 字符集指针。可以使用的 DTMF 字符包括：’0’、’1’、’2’、’3’、’4’、’5’、’6’、’7’、’8’、’9’、’*’、’#’、’a’、’b’、’c’、’d’

返回值：

-1	调用失败，失败原因为参数错误，或通道不支持本操作，具体失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用失败，失败原因为命令无法传递到板卡，具体原因可通过调用函数SsmGetLastErrMsg获得
1	调用成功

功能描述：

设置终止放音任务的 DTMF 字符集。

注意事项：

- 配置项DtmfStopPlayCharSet也可以实现相同功能，缺省的DTMF字符集为全集。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDtmfStopPlay，SsmGetDTMFStopPlayCharSet

2.9.3.3 SsmGetDtmfStopPlayFlag

获取放音任务是否因 DTMF 检测器检测到 DTMF 字符而终止的标志。

函数原型：

int SsmGetDtmfStopPlayFlag(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	标志为关闭状态
1	标志为开启状态

功能描述：

获取放音任务是否因 DTMF 检测器检测到 DTMF 字符而终止的标志。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDtmfStopPlay

2.9.3.4 SsmGetDTMFStopPlayCharSet

获取终止放音任务的 DTMF 字符集。

函数原型：

int SsmGetDTMFStopPlayCharSet(int ch, LPSTR lpstrDtmfCharSet)

参数说明：

ch	通道编号
lpstrDtmfCharSet	返回预设的 DMTF 字符集。

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg 获得
≥0	字符集长度

功能描述：

获取终止放音任务的 DTMF 字符集。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDTMFStopPlayCharSet

2.9.3.5 SsmSetBargeinStopPlay

设置放音任务是否因Barge in检测器检测到语音活动而终止。

函数原型：

int SsmSetBargeinStopPlay(int ch, BOOL bTerminatePlaybackOnBargein)

参数说明：

ch	通道编号
bTerminatePlaybackOnBargein	使能标志，取值范围： TRUE： 开启 FALSE：关闭

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置放音任务是否因Barge in检测器检测到语音活动而终止。

如果参数bTerminatePlaybackOnBargein设置为TRUE，当通道上正在执行放音任务时，如果Barge in检测器在来话信号中检测到了语音活动，驱动程序就会立即终止放音任务。

注意事项：

- 配置项DefaultBargeInStopPlay也可以实现相同功能，缺省值为关闭；
- 本函数设置的标志对预播放加载文件、单个文件、文件序列、单个内存、乒乓内存和内存序列均起作用。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetBargeinStopPlayFlag

2.9.3.6 SsmGetBargeinStopPlayFlag

获取放音任务是否因Barge in检测器检测到语音活动而终止的标志。

函数原型：

int SsmGetBargeinStopPlayFlag(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	标志为关闭状态
1	标志为开启状态

功能描述：

获取放音任务是否因Barge in检测器检测到语音活动而终止的标志。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetBargeinStopPlay

2.9.3.7 SsmSetHangupStopPlayFlag

设置放音任务是否因驱动程序状态机检测到对端挂机而终止。

函数原型：

int SsmSetHangupStopPlayFlag(int ch, BOOL bHangupStopPlayFlag)

参数说明：

ch	通道号
bHangupStopPlayFlag	驱动程序检测到对端挂机时，是否终止放音任务的标志，取值范围： TRUE： 开启此项功能 FALSE： 关闭此项功能

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置放音任务是否因驱动程序状态机在通道 Ch 上检测到对端挂机而终止。

注意事项：

- 配置项HangupStopPlay也可以实现相同功能，缺省值为关闭。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.9.4 播放单个文件的专用函数

2.9.4.1 SsmPlayFile

播放单个文件，启动文件播放任务。

函数原型：

int SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen)

参数说明：

ch	通道编号
pszFileName	语音文件名。支持标准的 wav 文件和无格式文件。如果是标准的 wav 文件，语音编码格式从文件头中获取，nFormat 参数将被忽略。如果是无格式文件，语音编码格式由 nFormat 参数指定。
nFormat	语音数据的编码格式，取值范围及其含义如下表所示： 取值 / 编码格式 / 备注 -2 PCM16 1 PCM8 6 A-Law 7 μ-Law 17 IMA ADPCM 要求板卡具有硬件解码能力 23 VOX 49 GSM 85 MP3 131 GC8 65411 G.729A 具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“板卡支持的CODEC”部分内容。
dwStartPos	文件中从实际的语音数据部分开始计算的语音片段的起始位置。对于标准的 wav 文件，起始位置为 0 是指文件头后的第 1 个语音数据字节；对于无格式文件，是从文件的头部开始计算。
dwLen	播放语音数据的长度（字节数）。如果本参数大于 dwStartPos 位置到文件尾的长度，则使用后者的长度作为实际的放音长度

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

播放单个文件。

调用本函数后，驱动程序内部会启动一个文件播放任务，当发生下列事件时，文件播放任务终止：

- ✧ 当文件中的全部数据播放完毕；
- ✧ 应用程序调用SsmStopPlayFile或SsmStopPlay；
- ✧ 驱动程序检测到DTMF、Barge in或对端挂机。详细内容请参见第 1 章中设置播放任务的终止条件部分。
- ✧ SHD系列的某些型号板卡在LinkFromStopPlayAndTone配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见第 1 章中“SHD系列板卡的原理框图”部分。

文件播放任务终止后，驱动程序会指定关闭文件，并向应用程序抛出E_PROC_PlayEnd事件。函数SsmCheckPlay也可以用来查询文件播放任务的完成情况。

在文件播放任务的执行期间，驱动程序会每隔一定时间向应用程序抛出E_PROC_PlayFile事件。驱动程序是否抛出此事件及抛出事件的时间间隔可以通过函数SsmSetEvent（携带E_PROC_PlayFile参数）设置，缺省值为每隔1000ms抛出事件，输出播放的时间长度。

在文件播放任务进行期间，应用程序可以调用下列函数来暂停（挂起）或调整放音位置：

- ✧ SsmPausePlay
- ✧ SsmFastFwdPlay
- ✧ SsmFastBwdPlay
- ✧ SsmSetPlayTime
- ✧ SsmSetPlayPrct

也可以调用下列函数或获取相关信息：

- ✧ SsmGetDataBytesToPlay
- ✧ SsmGetPlayedTime
- ✧ SsmGetPlayedPercentage

注意事项：

- 文件是以“共享读取”方式打开，因此本函数支持“边录边放”的功能；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopPlayFile，SsmStopPlay，SsmCheckPlay，SsmPausePlay，SsmRestartPlay，SsmFastFwdPlay，SsmFastBwdPlay，SsmSetPlayTime，SsmSetPlayPrct，SsmGetDataBytesToPlay，SsmGetPlayedTime，SsmGetPlayedPercentage

2.9.4.2 SsmStopPlayFile

终止文件播放任务。

函数原型：

int SsmStopPlayFile(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止调用SsmPlayFile函数启动的文件播放任务。

注意事项：

- 本函数可以用SsmStopPlay函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayFile，SsmPausePlay，SsmRestartPlay，SsmFastFwdPlay，SsmFastFwdPlay，SsmSetPlayTime，SsmSetPlayPrct，SsmStopPlay

2.9.4.3 SsmPausePlay

暂停（挂起）文件播放任务。

函数原型：

int SsmPausePlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

暂停（挂起）文件播放任务。文件播放任务暂停后，应用程序只能执行下列操作：

- 调用函数SsmStopPlayFile或SsmStopPlay终止文件播放任务；
- 调用函数SsmRestartPlay恢复文件播放任务。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

SsmRestartPlay，SsmStopPlayFile，SsmStopPlay

2.9.4.4 SsmRestartPlay

恢复因为调用函数SsmPausePlay而暂停的文件播放任务。

函数原型：

int SsmRestartPlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

恢复因为调用函数SsmPausePlay而暂停的文件播放任务。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPausePlay，SsmStopPlayFile，SsmStopPlay

2.9.4.5 SsmFastFwdPlay

跳过 1 秒钟的语音数据后继续播放。

函数原型：

int SsmFastFwdPlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

从当前放音位置开始，跳过 1 秒钟（可通过配置项FastPlayTime进行设置）的数据，然后重新启动单个文件放音任务。

注意事项：

- 如果文件中剩余的语音数据不足 1 秒，驱动程序视为全部语音数据播放完毕，将终止本次播放任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFastBwdPlay，SsmStopPlayFile，SsmStopPlay

2.9.4.6 SsmFastBwdPlay

回退 1 秒钟的语音数据后继续播放。

函数原型：

int SsmFastBwdPlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

从当前放音位置开始，回退 1 秒钟（可通过配置项FastPlayTime进行设置）的数据，然后继续文件播放任务。

注意事项：

- 如果已经回退到了文件首部，则从文件首部开始播放。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFastFwdPlay，SsmPausePlay，SsmStopPlayFile，SsmStopPlay

2.9.4.7 SsmSetPlayTime

跳转到文件中的指定时间位置后继续播放。

函数原型：

int SsmSetPlayTime(int ch, DWORD dwTime)

参数说明：

ch	通道编号
dwTime	以时间作为基准的新的播放位置，单位为毫秒。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

跳转到文件中的指定时间位置后继续播放。函数SsmGetPlayingFileInfo可以用来获取文件的持续时间等信息。

注意事项：

- 如果参数 dwTime 所对应的文件偏移量超过语音数据的长度，驱动程序视为全部语音数据播放完毕，将终止本次播放任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPlayingFileInfo，SsmSetPlayPrct，SsmStopPlayFile，SsmStopPlay

2.9.4.8 SsmSetPlayPrct

跳转到文件中的指定百分比位置后继续播放。

函数原型：

int SsmSetPlayPrct(int ch, DWORD dwPercentage)

参数说明：

ch	通道编号
dwPercentage	以占整个文件的百分比作为基准的新的放音起始位置，取值范围：0～100。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

跳转到文件中的指定百分比位置后继续播放，新的位置以百分比方式提供。函数SsmGetPlayingFileInfo可以用来获取文件的时间信息。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetPlayTime，SsmStopPlayFile，SsmStopPlay

2.9.4.9 SsmGetDataBytesToPlay

查询尚未播放的语音数据字节数。

函数原型：

int SsmGetDataBytesToPlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	尚未完成播放的语音数据长度，单位为字节

功能描述：

查询尚未播放的语音数据字节数。

注意事项：

- 本函数只支持由函数SsmPlayFile启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetDataBytesPlayed，SsmGetPlayedTime，SsmGetPlayedPercentage

2.9.4.10 SsmGetDataBytesPlayed

查询已经完成播放的语音数据长度。

函数原型：

int SsmGetDataBytesPlayed(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败

≥0	已经完成播放的语音数据长度，单位为字节

功能描述：

查询已经完成播放的语音数据长度。

注意事项：

- 本函数只支持由函数SsmPlayFile启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetDataBytesToPlay，SsmGetPlayedTime，SsmGetPlayedPercentage

2.9.4.11SsmGetPlayedTimeEx

获得从文件首部的第 1 个语音数据字节开始，到文件中的当前放音位置的播放持续时间。

函数原型：

int SsmGetPlayedTimeEx(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
≥0	返回从文件首部的第 1 个语音数据字节开始，到文件中的当前放音位置的播放持续时间，单位为毫秒。

功能描述：

获得从文件首部的第 1 个语音数据字节开始，到文件中的当前放音位置的播放持续时间。

例如，调用 SsmPlayFile(ch, “Voice.voc”, 7, 16000,)启动文件播放任务后，当驱动程序从文件偏移量 16000 处开始播放，并且播放完 10 秒的语音数据后，调用本函数，得到的返回值为 10＋2=12 秒（16000 字节μ率格式语音数据的时长为 2 秒）。

注意事项：

- 本函数只适用于SsmPlayFile启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPlayedTime

2.9.4.12SsmGetPlayingFileInfo

获取正在播放的放音文件的相关信息。

函数原型：

int SsmGetPlayingFileInfo(int ch , int *pnFormat , long *pnTime)

参数说明：

ch	通道编号
pnFormat	返回正在播放的文件的语音编码格式。有关SynCTI驱动平台中语音编码/解码格式的编码

	值请参见第 1 章中SynCTI支持的CODEC部分。
pnTime	文件中语音数据的总时长，单位毫秒。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

应用程序调用SsmPlayFile函数后，本函数用于获取正在播放的放音文件的相关信息。

注意事项：

- 本函数只适用于SsmPlayFile启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayFile，SsmGetPlayedTime，SsmGetPlayedTimeEx，SsmGetPlayedPercentage

2.9.5 播放文件序列的函数

2.9.5.1 SsmClearFileList

关闭并清除已提交给驱动程序的全部语音文件。

函数原型：

int SsmClearFileList(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

关闭并清除已提交给驱动程序的全部语音文件。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAddToFileList

2.9.5.2 SsmAddToFileList

向驱动程序提交一个需要播放的语音文件。

函数原型：

int SsmAddToFileList(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen)

参数说明：

ch	通道编号

pszFileName	语音文件名，可以是标准的 wav 文件或无格式文件。如果是标准的 wav 文件，语音编码格式从文件头中获取，nFormat 参数将被忽略。如果是无格式文件，语音编码格式由 nFormat 参数指定。
nFormat	语音数据的编码格式，取值范围及其含义如下表所示： 取值 / 编码格式 / 备注 1 / PCM8 6 / A-Law 7 / μ-Law 17 / IMA ADPCM / 要求板卡具有硬件解码能力 23 / VOX 具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“板卡支持的CODEC”部分内容。
dwStartPos	文件中从实际的语音数据部分开始计算的起始偏移量。例如，本参数为 0，对于标准的 wav 文件，表示从文件头后的第 1 个语音数据字节开始播放；对于无格式文件，是从文件的头部开始播放。
dwLen	播放语音数据的长度（字节数）。如果本参数大于文件中从 dwStartPos 开始计算的实际字节数，驱动程序视为播放到文件结尾。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

向驱动程序提交一个需要播放的语音文件。完成文件的提交后，即可以调用函数SsmPlayFileList启动文件序列播放任务。

注意事项：

- 本函数调用成功后，提交的文件被以“共享读”方式打开，并且一直保持打开状态，直到调用函数SsmClearFileList才会被关闭；
- 本函数可以连续调用，以提交多个文件，但最多不能超过配置项MaxPlayFileList的设定值，并且，所有文件都应该具有相同的语音编码格式。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayFileList，SsmStopPlay，SsmClearFileList

2.9.5.3 SsmPlayFileList

启动文件序列播放任务。

函数原型：

int SsmPlayFileList(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

启动文件序列播放任务。应用程序调用本函数后，驱动程序内部就启动了一个文件序列播放任务，并开始播放第 1 个文件。每当完成一个文件的播放后，驱动程序都会向应用程序抛出 E_PROC_PlayFileList 事件，然后开始播

放下一个文件。

当发生下列事件时，驱动程序会终止文件播放任务：

- ✧ 当文件中的全部数据播放完毕；
- ✧ 应用程序调用SsmStopPlayFileList或SsmStopPlay；
- ✧ 驱动程序检测到DTMF、Barge in或对端挂机。详细内容请参见第 1 章中设置播放任务的终止条件部分。
- ✧ SHD系列的某些型号板卡在LinkFromStopPlayAndTone配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见本章中“SHD系列板卡的原理框图”部分。

文件播放任务终止后，驱动程序会并向应用程序抛出E_PROC_PlayEnd事件。函数SsmCheckPlay也可以用来查询文件播放任务的完成情况。

注意事项：

- 文件播放任务终止后，驱动程序不会关闭提交的文件；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopPlayFileList，SsmStopPlay，SsmCheckPlay

2.9.5.4 SsmStopPlayFileList

终止文件序列播放任务。

函数原型：

int SsmStopPlayFileList(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止文件序列播放任务，该任务必须是由函数SsmPlayFileList启动。

注意事项：

- 本函数可以用SsmStopPlay函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayFileList，SsmClearFileList，SsmStopPlay

2.9.6 播放单个内存的函数

2.9.6.1 SsmPlayMem

启动单个内存播放任务。

函数原型：

int SsmPlayMem(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, DWORD dwStartOffset, DWORD dwStopOffset)

参数说明：

ch	通道编号
nFormat	语音数据的编码格式，取值范围及其含义如下表所示： 取值 / 编码格式 / 备注 6 / A-Law / 7 / μ-Law / 17 / IMA ADPCM / 要求板卡具有硬件解码能力 具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“板卡支持的CODEC”部分内容。
pBuf	存放语音数据的缓冲区地址。
dwBufSize	存放语音数据的缓冲区大小，单位为字节，不能低于 768 字节。
dwStartOffset	起始放音位置（偏移量），单位为字节，从缓冲区首地址开始计算。
dwStopOffset	结束放音位置（偏移量），单位为字节，从缓冲区首地址开始计算。如果本参数小于缓冲区长度，当驱动程序内部的放音指针到达dwStopOffset指定的位置后，本次放音任务结束；否则，表示对该缓冲区进行循环播放，直到应用程序调用SsmStopPlayMem，本次放音任务才会终止。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

启动单个内存播放任务。

调用本函数后，驱动程序内部会启动一个单个内存播放任务，同时维护一个初始值等于 dwStartOffset 的放音指针（偏移量），用于从提交的缓冲区中读取数据。当发生下列事件时，单个内存播放任务终止：

- ✧ 如果 dwStopOffset 小于提交的缓冲区大小，当驱动程序内部的放音指针到达参数 dwStopOffset 指定的停止放音位置，表明缓冲区中的全部数据播放完毕；
- ✧ 应用程序调用SsmStopPlayMem或SsmStopPlay；
- ✧ 驱动程序检测到DTMF、Barge in或对端挂机。详细内容请参见第 1 章中设置播放任务的终止条件部分。
- ✧ SHD系列的某些型号板卡在LinkFromStopPlayAndTone配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见本章中“SHD系列板卡的原理框图”部分。

单个内存播放任务终止后，驱动程序会向应用程序抛出E_PROC_PlayEnd事件。函数SsmCheckPlay也可以用来查询单个内存播放任务的完成情况。

在单个内存播放任务的执行期间，每当满足设定的条件时，驱动程序都会向应用程序抛出E_PROC_PlayMem事件。E_PROC_PlayMem事件的输出条件可以通过函数SsmSetEvent进行设置，可以设定为下列 3 种方式之一：

- ✧ 驱动程序完成了指定时间长度的语音数据的播放；
- ✧ 驱动程序内部的放音指针越过缓冲区的 1/2 处；
- ✧ 驱动程序内部的放音指针到达缓冲末尾。

有关E_PROC_PlayMem事件的输出条件设置请参见第 1 章中“MESSAGE_INFO”部分内容。

函数SsmSetStopPlayOffset 可以用来重新设置放音停止位置，函数SsmStopPlayMem或SsmStopPlay可以用来终止单个内存播放任务，函数SsmGetPlayOffset可以用来获取驱动程序内部的放音指针。

注意事项：

- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopPlayMem，SsmStopPlay，SsmSetStopPlayOffset，SsmGetPlayOffset， SsmSetEvent

2.9.6.2 SsmStopPlayMem

终止由函数SsmPlayMem启动的单个内存播放任务。

函数原型：

int WINAPI SsmStopPlayMem(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止由函数SsmPlayMem启动的单个内存播放任务。

注意事项：

- 本函数调用成功后，驱动程序仍然会向应用程序抛出 E_PROC_PlayEnd 事件；
- 函数SsmStopPlay可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayMem，SsmSetStopPlayOffset，SsmStopPlay

2.9.6.3 SsmSetStopPlayOffset

重新设置缓冲区中停止播放的位置。

函数原型：

int SsmSetStopPlayOffset(int ch, DWORD dwStopPlayOffset)

参数说明：

ch	通道编号
dwStopPlayOffset	内存放音的停止偏移值

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

重新设置缓冲区中停止播放的位置。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPlayOffset

2.9.6.4 SsmGetPlayOffset

获取单个内存播放任务时，驱动程序内部的放音指针。

函数原型：

int SsmGetPlayOffset(int ch, DWORD* pdwPlayOffset)

参数说明：

ch	通道编号
pdwPlayOffset	返回驱动程序内部的放音指针，一个以缓冲区首地址为基准的偏移量，单位为字节。

返回值：

-1	调用失败
0	调用成功

功能描述：

获取单个内存播放任务时，驱动程序内部的放音指针。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetStopPlayOffset

2.9.7 播放指定内存序列的函数

2.9.7.1 SsmAddToPlayMemList

向驱动程序提交一个内存缓冲区。

函数原型：

int SsmAddToPlayMemList(LPBYTE pBuf, DWORD dwDataLen, int nFormat)

参数说明：

pBuf	指向语音数据缓冲区的指针。内存缓冲区由应用程序分配空间
dwDataLen	语音数据缓冲区的大小，单位为字节
nFormat	语音编码格式。取值范围： 6：A-率 7：μ-率 17：IMA ADPCM

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

向驱动程序提交一个内存缓冲区，缓冲区中是需要播放的语音数据。驱动程序为每次提交的缓冲区分配一个编号，第 1 次提交的缓冲区编号为 0，依此类推。

本函数可以连续多次调用，但最多不能超过配置项MaxPlayMemList设定的值。驱动程序在播放时，会按照内存缓冲区的提交顺序依次播放。

注意事项：

- 本函数提交的内存缓冲区并非针对某个特定的通道，所有通道都可以使用已经提交的缓冲区；
- 提交的所有缓冲区中都必须使用相同的语音编码格式；
- 驱动程序在执行本函数时，只是记录缓冲区的信息，不会将缓冲区中的数据拷贝到驱动程序内部进行缓存，因此，应用程序必须保证没有任何通道在执行内存序列播放任务时，才能释放已经提交的缓冲区；
- 如果数据是 ADPCM 格式，则需注意帧结构，帧数据混乱会导致杂音。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmClearPlayMemList，SsmPlayMemList

2.9.7.2 SsmClearPlayMemList

清除已经提交给驱动程序的全部缓冲区。

函数原型：

int SsmClearPlayMemList(void)

参数说明：无

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

清除已经提交给驱动程序的全部缓冲区。

注意事项：

- 在调用本函数以前，必须保证没有任何通道在执行内存序列播放任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmAddToPlayMemList

2.9.7.3 SsmPlayMemList

启动内存序列播放任务。

函数原型：

int SsmPlayMemList(int ch, PWORD pMemList, WORD wMemListLen)

参数说明：

ch	通道编号
pMemList	数组指针，里面存放需要播放的缓冲区编号。缓冲区编号必须预先通过函数

	SsmAddToPlayMemList提交给驱动程序
wMemListLen	数组长度。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

启动内存序列播放任务。

调用本函数后，驱动程序内部会启动一个内存序列播放任务，按照 pMemList 中指定的顺序进行播放。当发生下列事件时，内存序列播放任务终止：

- ✧ pMemList 中指定的缓冲区中的语音数据全部播放完毕；
- ✧ 应用程序调用SsmStopPlayMemList或SsmStopPlay；
- ✧ 驱动程序检测到DTMF、Barge in或对端挂机。详细内容请参见第 1 章中设置播放任务的终止条件部分。
- ✧ SHD系列的某些型号板卡在LinkFromStopPlayAndTone配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见本章中“SHD系列板卡的原理框图”部分。

内存序列播放任务终止后，驱动程序会向应用程序抛出E_PROC_PlayEnd事件。函数SsmCheckPlay也可以用来查询内存序列播放任务的完成情况。

注意事项：

- 如果通道不支持对调用SsmAddToPlayMemList函数提交缓冲区时指定的语音编码格式进行硬件解码，本函数会失败。通道是否支持对某种语音编码格式进行硬件解码，取决于它所在板卡的型号，更多信息请参见第 1 章中相关内容；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopPlayMemList，SsmStopPlay，SsmCheckPlay，SsmAddToPlayMemList

2.9.7.4 SsmStopPlayMemList

终止由函数SsmPlayMemList启动内存序列播放任务。

函数原型：

int SsmStopPlayMemList(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

终止由函数SsmPlayMemList启动内存序列播放任务。

注意事项：

- 本函数调用成功后，驱动程序仍然会向应用程序抛出 E_PROC_PlayEnd 事件；
- 本函数可以用SsmStopPlay函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayMemList，SsmStopPlay

2.9.8 播放预加载语音的函数（CTI 系列）

2.9.8.1 预加载语音信息的初始化函数

2.9.8.1.1 SsmSetMaxIdxSeg

设置预加载语音片段的总数。

函数原型：

int SsmSetMaxIdxSeg(WORD wMaxIdxSeg)

参数说明：

wMaxIdxSeg	最大索引片段数

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

设置预加载语音片段的总数。配置项MaxIndexSeg也可以实现相同功能。

注意事项：

- 调用本函数时，如果此前已经存在预加载的语音片段，它们都将被卸载。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetTotalIndexSeg

2.9.8.1.2 SsmGetTotalIndexSeg

获取预加载语音片段的总数。

函数原型：

int SsmGetTotalIndexSeg ()

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	总的索引片段数

功能描述：

获取预加载语音片段的总数。

注意事项：无。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetMaxIdxSeg

2.9.8.1.3 SsmLoadIndexData

将指定的语音片段加载到内存。

函数原型：

int SsmLoadIndexData(int nSegNo, LPSTR pAlias, int nCodec, LPSTR pVocFile, long lStartPos, long lLen)

参数说明：

nSegNo	为语音片段分配的编号，取值范围：0~N，N为设置的预加载语音片段总数，可以通过函数SsmGetTotalIndexSeg获得
pAlias	为语音片段分配的别名，长度不能超过 20 个字符。不能以数字‘0’～‘9’作为别名字符串的首字符。
nCodec	语音数据的编码格式，取值范围及其含义如下表所示： 取值 / 编码格式 / 备注 6 / A-Law 7 / μ-Law 17 / IMA ADPCM / 要求板卡具有硬件解码能力 23 / VOX / 要求板卡具有硬件解码能力 具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“板卡支持的CODEC”部分内容。
pVocFile	包含语音片段的文件名，可以是标准的 wav 格式文件，也可以是没有文件头的无格式文件。如果是标准的 wav 文件，参数 nCodec 指定的格式将被忽略；否则被视为无格式文件，编码格式由参数 nCodec 指定。
lStartPos	文件中从实际的语音数据部分开始计算的语音片段的起始位置。对于标准的 wav 文件，起始位置为 0 是指文件头后的第 1 个语音数据字节；对于无格式文件，是从文件的头部开始计算。
lLen	语音片段的长度，单位为字节。注意： ● 如果本参数大于 lStartPos 到文件尾的数据长度，则实际加载的语音片段长度为 lStartPos； ● 如果 lLen 设置为-1，表示从 lStartPos 装载到文件尾 语音片段的字节长度与时间长度之间的换算值取决于编码格式，相关内容请参见第 1 章中的“SynCTI支持的CODEC”部分。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

本函数将语音片段数据从指定文件加载到内存。语音片段的数据编码格式按照下列规则确定：

注意事项：

- 对每个预加载语音片段都需要调用一次本函数；
- 配置文件Index.ini中的配置节SegNo可以实现相同功能；

- 应用程序应保证所有的预加载语音片段都具有相同的语音编码格式。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFreeIndexData

2.9.8.1.4 SsmFreeIndexData

卸载已经加载到内存的语音片段。

函数原型：

int SsmFreeIndexData(int nSegNo)

参数说明：

nSegNo	为语音片段分配的编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

卸载已经加载到内存的语音片段。卸载后，nSegNo 可以用来加载新的语音片段数据。

注意事项：

- 驱动程序在执行卸载SsmCloseCti函数时，会自动释放全部预加载语音片段所占用的内存等资源。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmLoadIndexData

2.9.8.1.5 SsmLoadChIndexData

将语音片段数据从指定文件加载到相对应通道的内存中。

函数原型：

int WINAPI SsmLoadChIndexData(int ch, int nSegNo, LPSTR pAlias, int nCodec, LPSTR pVocFile, long lStartPos, long lLen)

参数说明：

ch	通道号
nSegNo	分配给内存索引语音片段的段号
pAlias	分配给内存索引语音片段的别名，别名长度不能超过 20 个字符。
nCodec	语音片段数据的编码格式，可选值：6（A 率）、7（μ 率）、17（IMA ADPCM）。
pVocFile	包含语音数据的文件名

lStartPos	语音片段数据在语音文件的起始位置（不包括文件头）
lLen	装载语音数据的长度（字节数）。如果本参数指定的长度大于 lStartPos 到文件尾的数据长度，则实际的装载长度为后者的长度；如果 lLen=-1，则表示从 lStartPos 装载到文件尾。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

本函数将语音片段数据从指定文件加载到相对应通道的内存中。语音片段的数据编码格式按照下列规则确定：

1、 如果 pVocFile 是标准 WAV 文件，则参数 nCodec 将被忽略；

2、 如果 pVocFile 不是标准 WAV 文件，则该文件将被视为无格式文件，其编码格式由参数 nCodec 指定。

注意事项：

- 驱动程序在执行卸载SsmCloseCti函数时，会自动释放全部预加载语音片段所占用的内存等资源。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFreeChIndexData

2.9.8.1.6 SsmFreeChIndexData

卸载通道上已经装载到内存的语音片段数据。

函数原型：

int WINAPI SsmFreeChIndexData(int ch, int nSegNo)

参数说明：

ch	通道号
nSegNo	索引片段号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

卸载通道上已经装载到内存的语音片段数据。卸载后，该段号可以用来装载新的语音片段数据。

注意事项：

- 驱动程序在执行卸载SsmCloseCti函数时，会自动释放全部预加载语音片段所占用的内存等资源。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmLoadChIndexData

2.9.8.2 播放预加载语音的函数

2.9.8.2.1 SsmPlayIndexString

2.9.8.2.2 SsmPlayIndexList

顺序播放指定的一个或多个预加载语音片段。

函数原型：

int SsmPlayIndexList(int ch, WORD wIdxListLen, PWORD pwIdxList)

int SsmPlayIndexString(int ch, LPSTR pszIdxStr)

参数说明：

ch	通道编号
wIdxListLen	需要播放的预加载语音片段的数量，取值范围：1～N-1，N由配置项MaxPlayIndexList设定。
pwIdxList	存放预加载语音片段编号的数组，必须具有相同的语音编码格式。
pszIdxStr	包含预加载语音片段信息的字符串，存放需要播放的语音片段的别名或编号的序列。每一个预加载语音片段都可以使用编号或者别名来表征，片段之间用‘，’隔开。pszIdxStr中指定的预加载语音片段的总数不能超过配置项MaxPlayIndexList的设定值。注意：pszIdxStr中指定的所有语音片段必须具备相同的编码格式。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

按照 pwIdxList 或 pszIdxStr 指定的内容播放预加载语音片段。

调用本函数后，驱动程序内部会启动一个预加载语音播放任务，当发生下列事件时，预加载语音播放任务终止：

- ✧ 参数 pwIdxList 或 pszIdxStr 指定的全部语音数据播放完毕；
- ✧ 应用程序调用SsmStopPlayIndex或SsmStopPlay；
- ✧ 驱动程序检测到DTMF、Barge in或对端挂机。详细内容请参见第 1 章中设置播放任务的终止条件部分。
- ✧ SHD系列的某些型号板卡在LinkFromStopPlayAndTone配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见本章中“SHD系列板卡的原理框图”部分。

预加载语音播放任务终止后，驱动程序会向应用程序抛出E_PROC_PlayEnd事件。函数SsmCheckPlay也可以用来播放任务的完成情况。

应用程序可以随时调用函数SsmStopPlayIndex或SsmStopPlay来终止播放任务。

注意事项：

- 本函数只支持对pwIdxList或pszIdxStr中语音数据编码格式具备硬件解码功能的通道，详细内容请参见函数SsmLoadIndexData的说明。
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopPlayIndex，SsmStopPlay，SsmCheckPlay，SsmLoadIndexData

2.9.8.2.3 SsmStopPlayIndex

停止播放预加载语音片段。

函数原型：

int SsmStopPlayIndex(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

停止播放预加载语音片段。播放预加载语音片段的任务终止后，驱动程序会将终止播放的原因设置为CHKPLAY_APPLICATION_END，并向应用程序抛出 E_PROC_PlayEnd 事件。

注意事项：

- 播放任务必须由函数SsmPlayIndexString或SsmPlayIndexList启动；
- 本函数可以用SsmStopPlay函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayIndexString，SsmPlayIndexList，SsmStopPlay

2.9.9 播放乒乓缓存的函数（CTI 系列）

2.9.9.1 SsmPlayMemBlock

向驱动程序提交一个缓冲区，并启动乒乓内存播放任务。

函数原型：

int SsmPlayMemBlock(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, PLAYMEMBLOCKHANDLER pfnCallback, PVOID pV)

参数说明：

ch	通道编号
nFormat	语音编码格式。取值范围： 6：A-率 7：μ-率 17：IMA ADPCM（要求板卡具有硬件解码能力）
pBuf	指向语音数据缓冲区的指针
dwBufSize	语音数据缓冲区的大小，单位为字节。dwBufSize 的最小值按照下面规则确定： ● 如果配置项RecordAndPlayUseAsIP设置为 0（缺省值），dwBufSize应不低于 768 字节； ● 如果RecordAndPlayUseAsIP设置为 1，对于A-率和μ-率，dwBufSize应不低于 192 字节（推荐值：Windows操作系统下应不低于 192 字节，Linux下应不低于 384 字

	节）；对于IMA ADPCM格式，缓冲区大小应不低于 768 字节
pfnCallback	回调函数指针，当驱动程序完成一个缓冲区的播放后会自动调用本参数设置的回调函数。NULL 表示不设置回调函数。回调函数的原型在 shpa3api.h 中声明为 typedef BOOL (*PLAYMEMBLOCKHANDLER) (int **ch**, int **nEndReason**, PUCHAR **pucBuf**, DWORD **dwStopOffset**, PVOID **pV**); 其中，**ch**：通道编号； **nEndReason**：结束原因。取值范围： 1：缓冲区播放完毕； 2：放音任务因检测到 DTMF 按键字符而终止； 3：放音任务因检测到 bargein 而终止； 4：放音任务被应用程序终止； 5：放音任务因检测到对端挂机而终止。 **pucBuf**：播放完毕的缓冲区指针； **dwStopOffset**：播放完毕时，驱动程序内部的放音指针在缓冲区中偏移量，单位为字节。 **pV**：应用程序在调用本函数时传递的指针
pV	PVOID 类型指针，通常用于应用程序向回调函数传递一些必要的数据，如数据结构等。这个指针将由驱动程序透明地传递给回调函数。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

向驱动程序提交一个缓冲区，并启动乒乓内存播放任务。有关乒乓内存播放模式的详细内容请参见第 1 章中播放乒乓内存部分。

首次调用本函数时，驱动程序内部会启动一个乒乓内存播放任务，并开始按照提交缓冲区的顺序进行播放。每当驱动程序完成一个缓冲区的播放后，会自动切换到另一个缓冲区并开始播放，同时调用应用程序设置的回调函数，将完成播放的缓冲区信息传递给应用程序。应用程序可以在回调函数中更新缓冲区中的语音数据，然后本函数再次将缓冲区提交给驱动程序。

当发生下列事件时，内存序列播放任务终止：

- ✧ 驱动程序完成全部已提交的缓冲区的播放；
- ✧ 应用程序调用SsmStopPlayMemBlock或SsmStopPlay；
- ✧ 驱动程序检测到DTMF、Barge in或对端挂机。详细内容请参见第 1 章中设置播放任务的终止条件部分。
- ✧ SHD系列的某些型号板卡在LinkFromStopPlayAndTone配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见本章中“SHD系列板卡的原理框图”部分。

函数SsmCheckPlay也可以用来查询内存序列播放任务的完成情况。

注意事项：

- 如果通道不支持对 nFormat 参数指定的编码格式进行硬件解码，本函数会失败。通道是否支持对某种语音编码格式进行硬件解码，取决于它所在板卡的型号，详细内容请参见第 1 章中相关内容；
- 在回调函数中，应用程序完成语音数据的更新和提交的代码执行时间应尽可能短，不能大于驱动程序的中断时间（8 毫秒），否则可能引起驱动程序出错；
- 应用程序可以随时调用函数SsmStopPlayMemBlock或SsmStopPlay以终止双缓冲内存放音操作，但不能在回调函数中调用上述函数；
- 乒乓内存播放任务终止后，驱动程序不会向应用程序抛出 E_PROC_PlayEnd 事件；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数： SsmStopPlayMemBlock，SsmStopPlay，SsmPlayMem，SsmPlayMemList

2.9.9.2 SsmStopPlayMemBlock

终止由SsmPlayMemBlock启动的乒乓内存播放任务。

函数原型：

int SsmStopPlayMemBlock(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止由函数SsmPlayMemBlock启动的乒乓内存播放任务。

注意事项：

- 本函数不能在调用函数SsmPlayMemBlock时设置的回调函数中调用；
- 本函数可以用SsmStopPlay函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPlayMemBlock，SsmStopPlay

2.9.10 终止播放任务的通用函数

2.9.10.1 SsmStopPlay

终止以任何模式启动的播放任务。

函数原型：

int SsmStopPlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止以任何模式启动的播放任务。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：

SsmStopPlayFile，SsmStopPlayFileList，SsmStopPlayMem，SsmStopPlayMemList，SsmStopPlayIndex，SsmStopPlayMemBlock

2.9.11 获取播放任务的相关信息

2.9.11.1 SsmQueryPlayFormat

查询通道是否支持指定的放音格式。

函数原型：

int SsmQueryPlayFormat (int ch, int nFormat)

参数说明：

Ch	通道编号
nFormat	放音格式，取值范围及详细内容请参见第 1 章的SynCTI支持的CODEC部分。

返回值：

0	不支持
1	支持，硬件解码方式
2	支持，软件解码方式

功能描述：

查询通道是否支持指定的放音格式。

注意事项：

- 本函数适用于任何类型的通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.9.11.2 SsmGetPlayType

查询通道上正在执行的放音任务类型。

函数原型：

int SsmGetPlayType(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	没有放音任务
1	正在播放单个文件
2	正在播放单个文件，但被应用程序暂停
3	正在播放文件序列
4	正在播放预加载语音
5	正在播放单个内存
6	正在播放内存序列
7	正在发送 FSK 数据（注：发送 FSK 数据会占用放音资源）

8	正在使用放音操作来实现不具备卡间交换总线的通道的语音交换
9	正在播放乒乓内存

功能描述：

查询通道上正在执行的放音任务类型。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.9.11.3 SsmCheckPlay

获取放音任务的完成情况。

函数原型：

int SsmCheckPlay(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败。
0	放音正在进行。
1	放音任务终止：全部数据播放完毕。
2	放音任务因检测到DTMF字符而终止。详细内容请参见第 1 章中设置播放任务的终止条件部分。
3	放音任务因检测到bargein而终止。详细内容请参见第 1 章中设置播放任务的终止条件部分。
4	放音任务因检测到对端挂机而终止。详细内容请参见第 1 章中设置播放任务的终止条件部分。
5	放音任务被应用程序终止。
6	放音任务被应用程序调用SsmPausePlay函数暂停。放音任务必须为调用SsmPlayFile函数启动的文件播放任务。
7	放音任务因通道上执行了下总线操作而终止。详细内容请参见第 1 章中设置播放任务的终止条件部分。
8	放音任务因网络中断而停止。当应用程序调用SsmPlayFile函数播放网络上其它计算机上的语音文件时，如果驱动程序因各种原因，比如网络连接断开等，无法读取文件时，才会返回此值。

功能描述：

获取放音任务的完成情况。

注意事项：

- 放音任务可以由 SsmPlayFile、SsmPlayFileList、SsmPlayMemList、SsmPlayIndexList、SsmPlayIndexString、SsmPlayMem、SsmPlayMemBlock 中任何一个函数启动。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmPlayFile、SsmPlayFileList、SsmPlayMemList、SsmPlayIndexList、SsmPlayIndexString、SsmPlayMem、SsmPlayMemBlock

相关事件：E_PROC_PlayEnd，E_PROC_PlayFile，E_PROC_PlayFileList，E_PROC_PlayMem

2.9.11.4 SsmGetPlayedPercentage

获取已经完成播放的数据长度占总长度的百分比。

函数原型：

int SsmGetPlayedPercentage(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	已播放完成的字节数占总字节数的百分比

功能描述：

获取已经完成播放的数据长度占总长度的百分比。

注意事项：

- 本函数支持由函数SsmPlayIndexList、SsmPlayIndexString、SsmPlayFile、SsmPlayFileList、SsmPlayMemList启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPlayedTime

2.9.11.5 SsmGetPlayedTime

获取本次放音任务持续的时间。

函数原型：

int SsmGetPlayedTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
≥0	放音任务持续的时间，单位为毫秒

功能描述：

获取本次放音任务持续的时间。

注意事项：

- 本函数支持由函数SsmPlayFile、SsmPlayFileList、SsmPlayMemList、SsmPlayIndexList、SsmPlayIndexString、SsmPlayMem启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPlayedPercentage，SsmGetDataBytesToPlay，SsmGetPlayedTimeEx，SsmGetPlayingFileInfo

2.10 有关录音操作的函数

2.10.1 设置录音操作的参数

2.10.1.1 设置信号源及其音量

2.10.1.1.1SsmSetRecVolume

设置进入录音混音器的来话信号的音量调节器的增益。

函数原型：

int SsmSetRecVolume(int ch, int nVolume)

参数说明：

ch	通道编号
nVolume	录音音量，取值范围：-7~+6，-7 表示关闭，其余取值表示音量增益（乘以 3 即为分贝），大于 0 表示音量提升，小于 0 表示音量衰减。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置进入录音混音器 M3 的来话信号的音量调节器 A3 的增益。有关 M3 和 A3 的详细内容请参见第 1 章中相应板卡的原理框图部分。

注意事项：

- 通过配置项DefaultRecordVolume可以实现相同功能，缺省值为 0；
- 本函数设置的参数适用于由函数SsmPlayFile、SsmPlayFileList、SsmPlayMemList、SsmPlayIndexList、SsmPlayIndexString、SsmPlayMem中的任何一个函数启动的放音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecMixer，SsmSetRecBack

2.10.1.1.2SsmSetRecMixer

设置进入录音混音器 M3 的其它信号源的音量调节器 A2 的增益。

函数原型：

int SsmSetRecMixer(int ch, BOOL bEnRecMixer, int nMixerVolume)

参数说明：

ch	通道编号
bEnRecMixer	=FALSE：关闭音量调节器 A2，录音操作的数据源仅为本通道来话信号； =TRUE： 开启音量调节器 A2，录音操作的数据源除了本通道来话信号外，还包括其它来源。
nMixerVolume	其它录音信号源的音量增益，仅当 bEnRecMixer 开关为开启时有效。取值范围：-7~+6，其中， -7： 关闭 A2，相当于将 bEnRecMixer 参数设置为 FALSE； -6~6： 音量增益（乘以 3 即为分贝），大于 0 表示提升，小于 0 表示衰减。

返回值：

-1	调用失败
0	调用成功

功能描述：

设置进入录音混音器 M3 的其它信号源的音量调节器 A2 的增益。有关 M3 和 A2 的详细内容请参见第 1 章中相应板卡的原理框图部分。

注意事项：

- 配置项DefaultRecordMixerVolume可以实现相同功能，缺省为关闭A2；
- 本函数设置的参数所有类型的录音函数都起作用；
- 本函数不支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecVolume，SsmSetRecBack，SsmGetRecMixerState，SsmQueryOpRecMixer

2.10.1.1.3 SsmQueryOpRecMixer

查询通道是否支持设置音量调节器 A2 的增益的操作。

函数原型：

int SsmQueryOpRecMixer(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	不支持
1	支持

功能描述：

查询通道是否支持设置音量调节器 A2 的增益的操作。

注意事项：

- 本函数不支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecMixer

2.10.1.1.4SsmGetRecMixerState

获取音量调节器 A2 的增益参数设置。

函数原型：

int SsmGetRecMixerState(int ch, int* pnEnRecMixer, int* pnMixerVolume)

参数说明：

ch	通道编号
pnEnRecMixer	存放录音混音器开关状态的指针
pnMixerVolume	存放录音混音器中本通道来话信号外的另一路声音信号的音量。得到的音量值乘以 3 即为分贝

返回值：

-1	调用失败
0	调用成功

功能描述：

获取音量调节器 A2 的增益参数设置。

取得指定通道上的录音混音器的状态。如果录音混音器开关状态为开启，参数 pnEnRecMixer 将被置为 1，否则为 0。参数 pnMixerVolume 用于返回加入本通道来话信号混音的另一路声音的音量。

注意事项：

- 本函数设置的录音混音器开关对文件录音和内存录音函数都有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecMixer，SsmQueryOpRecMixer

2.10.1.1.5DTRSetMixerVolume

设置 DST 系列板卡上来话/去话音量调节器 A4-1、A4-2 的增益。

函数原型：

int DTRSetMixerVolume(int ch, int nGroup, int nInboundGain, int nOutboundGain)

参数说明：

ch	通道号
nGroup	选择 B 信道。数字电话机为 2B+D 模式，由于目前驱动程序不支持对 2 个 B 信道同时进行处理，本参数必须设置为 0
nInboundGain	来话信号音量调节器（即第 1 章中DST系列板卡的工作原理框图中的音量调节器A4-1）的增益，取值范围-6~+6，负值为衰减，正值为提升。本参数乘以 3 即为分贝值。
nOutboundGain	去话信号音量调节器（即第 1 章中DST系列板卡的工作原理框图中的音量调节器A4-2）的增益，取值范围-6~+6，负值为衰减，正值为提升。本参数乘以 3 即为分贝值。

返回值：

-1	调用失败
0	调用成功

功能描述：

设置DST系列板卡上来话/去话音量调节器A4-1、A4-2 的增益。有关A4-1、A4-2 的详细内容请参见第 1 章中DST系列板卡的工作原理框图部分。

注意事项：

- 本函数只支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecVolume，DTRGetMixerVolume

2.10.1.1.6DTRGetMixerVolume

获取 DST 系列板卡上来话/去话音量调节器 A4-1、A4-2 的增益参数。

函数原型：

int DTRGetMixerVolume(int ch, int nGroup, int* pnInboundGain, int* pnOutboundGain)

参数说明：

ch	通道号
nGroup	选择 B 信道，必须设置为 0
pnInboundGain	返回来话信号音量调节器（即第 1 章中DST系列板卡的工作原理框图中的音量调节器 A4-1）的增益参数，取值范围-6~+6，负值为衰减，正值为提升。本参数乘以 3 即为分贝值。
pnOutboundGain	返回去话信号音量调节器（即第 1 章中DST系列板卡的工作原理框图中的音量调节器 A4-2）的增益参数，取值范围-6~+6，负值为衰减，正值为提升。本参数乘以 3 即为分贝值。

返回值：

-1	调用失败
0	调用成功

功能描述：

获取DST系列板卡上来话/去话音量调节器A4-1、A4-2 的增益参数。有关A4-1、A4-2 的详细内容请参见第 1 章中DST系列板卡的工作原理框图部分。

注意事项：

- 本函数只支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：DTRSetMixerVolume

2.10.1.1.7 SsmSetRecBack

设置是否将总线混音器的输出、放音数据作为录音混音器的输入信号源。

函数原型：

int SsmSetRecBack(int ch, int nSelector)

参数说明：

ch	通道编号
nSelector	设置开关 k6-2 和 k6-1 的状态。取值为： 0：k6-2 断开，k6-1 断开 1：k6-2 闭合，k6-1 闭合 2：k6-2 断开，k6-1 闭合 3：k6-2 闭合，k6-1 断开 有关这 2 个开关的更多信息请参见的 2 章中SHD系列板卡的工作原理框图部分。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置是否将总线混音器的输出、放音数据作为录音混音器的输入信号源。

注意事项：

- 本函数只支持 SHD、SHN 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecVolume，SsmSetRecMixer

2.10.1.1.8 SsmSetNoModuleChBusRec

将没有安装任何业务模块的 ATP 系列板卡上的通道作为一个具有录音功能的资源通道使用。

函数原型：

int SsmSetNoModuleChBusRec(int ch, int nUsedAsResourceRecCh)

参数说明：

ch	通道编号
nUsedAsResourceReCh	1：启用此功能 0：不使用此功能

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

将没有安装任何业务模块的ATP系列板卡上的通道作为一个具有录音功能的资源通道使用。详细内容请参见第 1 章中ATP系列板卡的工作原理框图的“未安装业务模块的通道的特殊应用”部分。

注意事项：

- 启动录音前，应该先调用函数SsmSetRecVolume或设置配置项DefaultRecordVolume，以关闭来话信号的音量调节器；
- 录音的信号源只能来自 TDM 交换总线；
- 配置项NoModuleChBusRec可以实现相同功能；
- 本函数只支持 ATP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecVolume

2.10.1.2 设置录音任务的终止条件

2.10.1.2.1 SsmSetDTMFStopRecCharSet

设置录音任务是否因DTMF检测器检测到DTMF字符而终止。

函数原型：

int SsmSetDTMFStopRecCharSet(int ch, LPSTR lpstrDtmfCharSet)

参数说明：

ch	通道号
lpstrDtmfCharSet	ASCII 形式的终止录音任务的 DTMF 字符集指针。可以使用的 DTMF 字符包括：’0’、’1’、’2’、’3’、’4’、’5’、’6’、’7’、’8’、’9’、’*’、’#’、’a’、’b’、’c’、’d’，也可以设置为空，表示不使用此项功能。

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

设置录音任务是否因DTMF检测器检测到DTMF字符而终止。

如果参数lpstrDtmfCharSet不为空，当通道上正在执行录音任务时，如果DTMF检测器在来话信号中检测到了DTMF字符，并且该DTMF字符包含在用lpstrDtmfCharSet参数指定的DTMF字符集中，驱动程序就会立即终止录音任务。

注意事项：

- 配置项DtmfStopRecCharSet可以实现相同功能；
- 由文件录音函数、内存录音函数、乒乓内存录音函数启动的录音任务均支持此项功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetHangupStopRecFlag，SsmGetDTMFStopRecCharSet

2.10.1.2.2SsmGetDTMFStopRecCharSet

查询录音任务是否因DTMF检测器检测到DTMF字符而终止。

函数原型：

int SsmGetDTMFStopRecCharSet(int ch, LPSTR lpstrDtmfCharSet)

参数说明：

ch	通道号
lpstrDtmfCharSet	返回 ASCII 形式的 DTMF 字符集。

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	此项功能被关闭
>0	预设的 DTMF 字符集中字符的个数。

功能描述：

查询录音任务是否因DTMF检测器检测到DTMF字符而终止。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDTMFStopRecCharSet

2.10.1.2.3SsmSetHangupStopRecFlag

设置录音任务是否因驱动程序状态机检测到对端挂机而终止。

函数原型：

int SsmSetHangupStopRecFlag(int ch, BOOL bHangupStopRecFlag)

参数说明：

ch	通道号
bHangupStopRecFlag	驱动程序检测到对端挂机时，是否终止放音任务的标志。取值范围： TRUE：　开启此项功能 FALSE：　关闭此项功能

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

设置录音任务是否因驱动程序状态机检测到对端挂机而终止。

注意事项：

- 配置项HangupStopRec可以实现相同功能。驱动程序启动时，此功能为关闭状态。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDTMFStopRecCharSet

2.10.1.3设置预录音功能（REC 系列）

2.10.1.3.1SsmSetPrerecord

设置预录音功能。

函数原型：

int SsmSetPrerecord(int ch, BOOL bEnable, int nMode, WORD wInsertTime, int nFormat)

参数说明：

ch	通道编号
bEnable	TRUE： 开启预录音功能 FALSE：关闭预录音功能 **注：**本参数对应的配置项为PrerecordEnable
nMode	设置预录音功能的工作模式： 0： 当 Barge in 检测器检测到 Barge in 时开始将录音数据写入到内部缓冲区 1： 当模拟录音通道在线路上检测到摘机动作时开始将录音数据写入到内部缓冲区 **注：**本参数对应的配置项为PrerecordMode
nInsertTime	预录音功能启动后，当应用程序启动文件录音任务时，本参数设置将预录音的数据写入到文件中的时间长度，单位为毫秒。 **注：**本参数对应的配置项为PrerecordInsertTime
nFormat	设置录音的编码格式，取值范围： 6：A-率 7：μ-率 17：IMA ADPCM（注：要求板卡具有硬件编码器） **注：**本参数对应的配置项为PrerecordCodec

返回值：

-1	调用失败
0	调用成功

功能描述：

设置预录音功能。有关预录音功能的详细内容请参见第 1 章中设置预录音功能部分。

注意事项：

- 本函数只适用于 ATP 系列板卡；
- 预录音功能只支持文件录音函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPrerecordState

2.10.1.3.2SsmGetPrerecordState

获取预录音功能的设置。

函数原型：

int SsmGetPrerecordState (int ch , int* pnMode, PWORD pwInsertTime, int* pnFormat)

参数说明：

ch	通道编号
pnMode	返回预录音的工作模式
pnInsertTime	返回预录音数据写入到文件中的时间长度，单位为毫秒
pnFormat	返回预录音操作的录音格式

返回值：

-1	调用失败
0	预录音功能为关闭状态
1	预录音功能为开启状态。只有返回值为 1 时参数表中返回的值才有意义

功能描述：

获取预录音功能的设置。

注意事项：

相关参数的含义请参见SsmSetPrerecord函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetPrerecord

2.10.1.4 设置录音文件截尾功能

2.10.1.4.1 SsmSetTruncateTail

设置录音文件截尾功能的工作参数。

函数原型：

int SsmSetTruncateTail(int ch, DWORD dwTime)

参数说明：

ch	通道编号
dwTime	设置需要抛弃的录音数据长度，单位为毫秒。0 表示关闭此项功能。 对于 SynCTI Ver. 3.4.x，在进行截尾操作时，截尾时间长度为终止文件录音任务的 MF 字符的持续时间；对于 Ver. 3.5 及以后版本，如果 dwTime 大于终止文件录音任务的 DTMF 字符的持续时间，被截断的数据长度为 dwTime；否则为该 DTMF 的实际持续时间。 本参数也可以通过配置项TruncateTailOnRecordToFile进行设置。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置录音文件截尾功能的工作参数。有关录音文件截尾功能的详细内容请参见第 1 章中设置录音文件截尾功能部分。

文件录音任务启动后，如果开启了录音文件截尾功能和DTMF字符终止文件录音任务功能，，当DTMF检测器检测到了DTMF字符时，如果该DTMF字符落在函数SsmSetDTMFStopRecCharSet或配置项DtmfStopRecCharSet设定

的字符集中，驱动程序会立即终止文件录音任务，并将文件尾部的一段语音数据删除，被删除的语音数据长度由本函数设置。

注意事项：

- 如果文件录音操作为正常终止，即完成了指定长度数据的录制，驱动程序不会进行截尾操作；
- 录音文件截尾功能只支持文件录音函数，并且必须与"检测到 DTMF 时自动终止文件录音任务"的功能配合使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecToFile，SsmRecToFileA，SsmRecToFileB，SsmRecToFileEx ，SsmGetTruncateTailTime

2.10.1.4.2 SsmGetTruncateTailTime

获取录音文件截尾功能的工作参数。

函数原型：

long SsmGetTruncateTailTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	此项功能未开启
>0	返回设置的截尾时间参数，单位为毫秒

功能描述：

获取录音文件截尾功能的工作参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetTruncateTail

2.10.1.5 设置录音 AGC

2.10.1.5.1 SsmSetRecAGC

开启或关闭 AGC 功能。

函数原型：

int SsmSetRecAGC(int ch, BOOL bEnable)

参数说明：

ch	通道号
bEnable	录音自动增益控制开关（AGC）。 =TRUE：打开； =FALSE：关闭

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

开启或关闭自动增益控制（Automatic Gain Control,AGC）功能。如果开启 AGC 功能，驱动程序会根据输入信号的幅度对信号自动进行调节，对小信号进行提升，对大信号进行衰减。

注意事项：

- 本函数设置的录音 AGC 开关对任何类型的录音操作均有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRecAGCSwitch

2.10.1.5.2 SsmGetRecAGCSwitch

获取录音 AGC 的工作状态。

函数原型：

int SsmGetRecAGCSwitch(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	AGC 为关闭状态
1	AGC 为开启状态

功能描述：

获取指定通道上 AGC 模块的工作状态。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRecAGC

2.10.2 获取录音操作的信息

2.10.2.1 SsmQueryRecFormat

查询通道是否支持指定的录音格式。

函数原型：

int SsmQueryRecFormat (int ch, int nFormat)

参数说明：

ch	通道编号
nFormat	录音格式

返回值：

0	不支持该格式
1	支持该格式，硬编码
2	支持该格式，软编码

功能描述：

查询通道是否支持指定的录音格式。

注意事项：

- 本函数支持所有的板卡和通道类型。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.10.2.2 SsmGetRecType

获取正在执行的录音任务类型。

函数原型：

int SsmGetRecType(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	通道上没有录音任务
1	正在执行文件录音任务
2	文件录音任务被暂停
3	正在执行内存录音任务
4	正在执行乒乓缓冲区录音任务

功能描述：

获取正在执行的录音任务类型。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.10.2.3 SsmGetRecTime

获取录音任务持续的时间长度。

函数原型：

int SsmGetRecTime(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	录音任务持续的时间，单位为毫秒

功能描述：

获取录音任务持续的时间。

注意事项：

- 本函数只支持文件录音任务和内存录音任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecToFile，SsmRecToFileA，SsmRecToFileB，SsmRecToFileEx ，SsmRecToMem

2.10.2.4 SsmCheckRecord

查询录音任务的完成情况。

函数原型：

int SsmCheckRecord (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	录音任务进行中
1	录音任务被应用程序终止
2	录音任务因检测到 DTMF 按键字符而终止
3	录音操作因检测到对端挂机而终止
4	录音任务自然终止
5	文件录音任务被暂停
6	文件录音任务因写入文件失败而终止

功能描述：

查询录音任务的完成情况。

注意事项：

- 本函数可适用于任何方式启动的录音任务，包括SsmRecToFile，SsmRecToFileA，SsmRecToFileB，SsmRecToFileEx ，SsmRecToMem， SsmRecordMemBlock。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相 关 函 数 ： SsmRecToFile， SsmRecToFileA， SsmRecToFileB， SsmRecToFileEx， SsmRecToMem， SsmRecordMemBlock

2.10.3 录制到文件的函数

2.10.3.1SsmRecToFile

2.10.3.2SsmRecToFileA

2.10.3.3SsmRecToFileB

2.10.3.4SsmRecToFileEx

2.10.3.5SpyRecToFile

启动指定通道上的文件录音任务。

SsmRecToFileA 除了具有 SsmRecToFile 的全部功能外，还提供了通过回调函数来获得缓存在驱动程序内存中的录音数据流的能力。

SsmRecToFileEx 除了具有 SsmRecToFile 的全部功能外，还提供利用 Barge in 检测器输出的 Bargin 信号来触发录音数据写入文件的功能。

SpyRecToFile 启动基于电路编号的文件录音任务，只适用于 DTP 系列板卡。

函数原型：

int **SsmRecToFile**(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask)

int **SsmRecToFileA**(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackA)

int **SsmRecToFileB**(int ch,LPSTR pszFileName,int nFormat,DWORD dwStartPos,DWORD dwBytes,DWORD dwTime,int nMask,LPRECTOMEMB pfnCallbackB, PVOID pVoid);

int **SsmRecToFileEx**(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, BOOL bSaveToFileOnBargin, DWORD dwRollbackTime)

int **SpyRecToFile**(int nCic, WORD wDirection, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask)

参数说明：

<table>
<tr><td>ch</td><td>通道编号</td></tr>
<tr><td>pszFileName</td><td>保存语音数据的文件名。
如果参数 pszFileName 声明的文件不存在，驱动程序按照下列规则自动创建一个新文件：
✧ 如果 pszFileName 以“.wav”作为文件扩展名，驱动程序认为应用程序希望录制标准 WAV 文件，因此会创建一个标准的 WAV 文件；
✧ 如果不是以“.wav”作为文件扩展名，驱动程序判断为无格式文件，文件中的数据全部是语音数据，没有文件头。</td></tr>
<tr><td>nFormat</td><td>录音数据的编码格式，取值范围如下表所示：
<table>
<tr><th>取值</th><th>CODEC</th><th>SsmRecToFile</th><th>SsmRecToFileA
SsmRecToFileB</th><th>SsmRecToFileEx</th><th>SpyRecToFile</th></tr>
<tr><td>-2</td><td>PCM16</td><td>☆</td><td>☆</td><td>—</td><td>☆</td></tr>
<tr><td>1</td><td>PCM8</td><td>☆</td><td>☆</td><td>—</td><td>☆</td></tr>
<tr><td>6</td><td>A-Law</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>7</td><td>μ-Law</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>17</td><td>IMA ADPCM</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>23</td><td>VOX</td><td>√</td><td>√</td><td>—</td><td>√</td></tr>
</table></td></tr>
</table>

	49 GSM ★ ★ — ★ 85 MP3 ◇ ◇ — ◇ 131 GC8 √ √ — √ 65411 G.729A √ √ — √ 图例： ☆ 使用软件编码器，由驱动程序通过软件编码实现 √ 使用硬件编码器实现，要求硬件具有编码器。更多信息请参见第 1 章“SynCTI支持的CODEC” ★ 软件编码，通过外挂 ACM 程序实现 — 不支持 ◇ 如果硬件支持，使用硬件编码器；否则使用软件编码器，通过外挂 ACM 程序实现 实际的录音编码格式按照下列规则确定： 如果参数 pszFileName 声明的文件已经存在，并且是标准的 wav 文件，则编码格式为 wav 文件头中指定的编码格式，本参数将被忽略； 如果已经存在的文件不是标准的 wav 文件，或者文件为新创建文件，编码格式由 nFormat 决定。 **注意：如果开启了预录音功能， nFormat 必须与预录音功能设置的语音编码格式相同，否则本函数会返回失败；**
dwStartPos	第 1 包录音数据写入到文件时的起始偏移量。如果目标文件具有 wav 文件头，dwStartPos 从语音数据区的第 1 个直接开始计算，不包括文件头长度。 ✧ 如果文件已经存在：如果 dwStartPos 小于或等于文件中的语音数据长度，从本参数指定的位置开始写入数据；如果 dwStartPos 大于文件中的语音数据长度，从文件尾部开始写入录音数据。 ✧ 如果是新创建文件：本参数将被忽略。如果是标准 wav 文件，紧随文件头开始写入；如果不是，从文件首部开始写入。
dwBytes	设置需要录制的语音数据的长度，单位为字节。 当录制完成的总字节数大于本参数设置的字节数时，录音任务结束。将本参数设置为一个很大的值（例如 0xffffffff）就可以达到持续录音的目的。 本参数是否起作用由 nMask 参数控制。 **注意：如果使用具有帧结构的语音编码格式（如 IMA ADPCM、GSM、MP3、G.729A 等），并且 dwBytes 不是帧长度的整数倍时，驱动程序会自动调整 dwBytes 为帧长度的整数倍，以保证帧结构的完整性。**
dwTime	设置需要录制的语音数据的总时长，单位为毫秒。 当录制任务持续的总时长大于本参数设置的时长时，录音任务结束。将本参数设置为一个很大的值（例如 0xffffffff）就可以达到持续录音的目的。 本参数是否起作用由 nMask 参数控制。 **注意：如果使用具有帧结构的语音编码格式（如 IMA ADPCM、GSM、MP3、G.729A 等），并且 dwTime 参数对应的字节数不是帧长度的整数倍时，驱动程序会自动调整 dwTime 为帧长度的整数倍，以保证帧结构的完整性。**
nMask	设置停止录音的方式。取值范围： 0：以 dwBytes 参数作为停止录音的条件。 1：以 dwTime 参数作为停止录音的条件。
pfnCallbackA pfnCallbackB	设置回调函数指针。如果pfnCallbackA或pfnCallbackB为NULL，SsmRecToFileA或SsmRecToFileB的功能与SsmRecToFile完全相同；如果不为NULL，当驱动程序的内部缓冲区积累了 16384 字节的录音数据并将其一次性写入文件后，会调用本参数设置的回调函数，将缓存的语音数据流同步传递给应用程序。函数SsmSetFlag（携带F_RECTOFILEA_CALLBACKTIME参数）可以用来设置回调的间隔。 回调函数 pfnCallbackA 的原型： void CALLBACK RecFileCallback(int **ch**, LPBYTE **lpData**, DWORD **dwDataLen**); 回调函数 pfnCallbackB 的原型： void CALLBACK RecFileCallback (int **ch**, LPBYTE **lpData**, DWORD **dwDataLen**, PVOID **pVoid**); 上式中， **ch**： 语音通道号 **lpData**： 驱动程序内部的录音数据缓冲区地址 **dwDataLen**：录音数据长度（字节数） **pVoid**： 调用 SsmRecToFileB 时传入的指。

pVoid	PVOID 类型的指针，用于将应用程序的对象指针透明地传输给回调函数 pfnCallbackB
bSaveToFileOnBargin	是否启用Barge in检测器的输出来触发将录音数据写入文件的功能。取值范围： TRUE： 启用 FALSE：不启用 如果本参数为 FALSE，dwRollbackTime 参数将被忽略，SsmRecToFileEx 函数与 SsmRecToFile 功能完全一致
dwRollbackTime	本参数用于决定将 Barge in 信号产生以前多少时间的语音数据保存到文件，单位为毫秒，最大值为 2000ms。仅当 bSaveToFileOnBargin 参数设置为 TRUE 时有效。 由于Barge in检测器检测到Barge in需要一定的时间，为了保证录制的语音数据的完整性，需要合理设置本参数。建议本参数应略大于函数SsmSetIsBargeInDtrmTime 或配置项BargeInDtrmTime所设置的参数。
nCic	SpyCic的逻辑编号。更多信息请参见第 1 章中“DTP系列板卡”部分内容。
wDirection	设置录音信号源，取值范围： 0：只录制主叫方的声音。驱动程序自动在与 nCic 绑定的主叫方通道上调用 SsmRecToFile 函数，因此，相关的输出消息也在主叫方通道上返回； 1：只录制被叫方的声音。驱动程序自动在与 nCic 绑定的被叫方通道上调用 SsmRecToFile 函数，因此，相关的输出消息也在被叫方通道上返回； 2：混合录制主叫方与被叫方的声音。驱动程序自动在与 nCic 绑定的被叫方通道上调用 SsmRecToFile 函数，因此，相关的输出消息也在被叫方通道上返回

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

启动指定通道上的文件录音任务。

SsmRecToFileA 和 SsmRecToFileB 除了具有 SsmRecToFile 的全部功能外，可以将保存在驱动程序内部缓冲区的录音数据流以回调函数的方式开放给应用程序。如果参数 pfnCallbackA（或 pfnCallbackB）为 NULL，SsmRecToFileA（或 SsmRecToFileB）的功能与 SsmRecToFile 函数完全相同。

SsmRecToFileEx 函数启动录音任务后，驱动程序只是将录音数据缓存到内部的录音缓冲区中，并不将录音数据写入到文件。只有当 Barge in 检测器检测到 Barge in 后，驱动程序才开始将内部缓冲中的录音数据写入到文件。如果在本次录音任务终止时仍然没有检测到 Bargin，则不会将任何数据写入到录音文件。

应用程序调用文件录音的函数后，驱动程序内部会启动一个文件录音任务，当发生下列事件时，文件录音任务终止：

- ✧ 文件录音任务自然终止，即驱动程序录制的语音数据长度到达参数 dwBytes 或 dwTime 参数指定的长度；
- ✧ 驱动程序将语音数据写入文件时失败；
- ✧ 应用程序调用SsmStopRecToFile；
- ✧ 驱动程序检测到DTMF或对端挂机，详细内容请参见第 1 章中设置录音任务的终止条件部分。

文件录音任务终止后，驱动程序会自动关闭文件，并向应用程序抛出E_PROC_RecordEnd事件。如果开启了录音文件截尾功能，驱动程序会自动将文件尾部的一段数据抛弃。有关录音文件截尾功能的详细内容请参见第 1 章中设置录音文件截尾功能部分。

在文件录音任务的执行期间，驱动程序会每隔 1000ms向应用程序抛出E_PROC_RecordFile事件。驱动程序是否抛出此事件及抛出事件的条件可以通过SsmSetEvent函数（携带E_PROC_RecordFile参数）设置，缺省值为每隔 1000 毫秒抛出此事件。函数SsmChkRecToFile或SsmCheckRecord也可以用来查询文件录音任务的完成情况。

在文件录音任务执行过程中，应用程序可以：

- ✧ 调用SsmPauseRecToFile函数暂停文件录音任务；
- ✧ 调用SsmStopRecToFile函数终止文件录音任务；
- ✧ 调用SsmChkRecToFile或SsmCheckRecord函数查询文件录音任务的完成情况；

- ✧ 调用SsmGetRecTime函数获取文件录音任务持续的时间；
- ✧ 调用SsmGetDataBytesToRecord函数获取尚未完成的录音数据字节数。

注意事项：

- 有关选择录音信号源、设置音量的详细内容请参见第 1 章中“设置录音信号源”部分内容；
- 如果开启了预录音功能，驱动程序会根据预录音的参数设置将保存在内部缓冲区的预录音数据先写入到文件，然后才开始执行正常的文件录音任务，将后续录音数据写入文件。有关预录音功能的详细内容请参见第 1 章中“设置预录音功能”部分内容；
- 如果nFormat为GSM或MP3，必须将配置项GsmCodecEnable设置为非 0，并保证Windows系统中已经安装了相应的音频编码解码器(ACM)驱动程序；
- SpyRecToFile启动基于电路编号的文件录音任务，只适用于DTP系列板卡。SpyRecToFile函数是基于电路编号的函数，它向应用程序抛出的消息是基于呼叫过程中与nCic绑定的主叫方或被叫方通道。函数SpyGetCallInCh、SpyGetCallOutCh分别用于获取本次呼叫中的被叫方和主叫方的通道编号。函数SpyStopRecToFile用于停止SpyRecToFile函数启动的文件录音任务。

相关信息：

驱动程序版本要求	SsmRecToFileB 要求的 SynCTI 版本为 Ver. 4.7.2.0 或更高； 其余函数要求 SynCTI Ver. 2.0 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopRecToFile，SsmChkRecToFile，SsmGetRecTime，SsmPauseRecToFile，SsmRestartRecToFile，SsmGetDataBytesToRecord，SsmSetDTMFStopRecCharSet，SsmSetHangupStopRecFlag，SsmSetRecVolume，SsmSetRecMixer，SsmSetRecBack，SsmSetRecAGC

2.10.3.6SsmStopRecToFile

2.10.3.7SpyStopRecToFile

终止用文件录音任务。SpyStopRecToFile 用于终止调用SpyRecToFile函数启动的文件录音任务。

函数原型：

int SsmStopRecToFile(int ch)

int SpyStopRecToFile(int nCic)

参数说明：

ch	通道编号
nCic	被监控电路编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

SsmStopRecToFile终止由函数SsmRecToFile、SsmRecToFileA、SsmRecToFileB或SsmRecToFileEx启动的文件录音任务，SpyStopRecToFile 用于终止调用SpyRecToFile函数启动的文件录音任务。

注意事项：

- 本函数调用后，驱动程序也会抛出 E_PROC_RecordEnd 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecToFile，SsmRecToFileA，SsmRecToFileB，SsmRecToFileEx，SsmSetDTMFStopRecCharSet，SsmSetHangupStopRecFlag，SsmSetRecVolume，SsmSetRecMixer，SsmSetRecBack

2.10.3.8 SsmPauseRecToFile

暂停文件录音任务。

函数原型：

int SsmPauseRecToFile(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

暂停由函数SsmRecToFile、SsmRecToFileA、SsmRecToFileB或SsmRecToFileEx启动的文件录音任务。文件录音任务被挂起后，只能通过函数SsmRestartRecToFile恢复，或者通过函数SsmStopRecToFile终止。

注意事项：

- 本函数只支持SsmRecToFile，SsmRecToFileA，SsmRecToFileB，SsmRecToFileEx启动的任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRestartRecToFile，SsmStopRecToFile

2.10.3.9 SsmRestartRecToFile

恢复文件录音任务。

函数原型：

int SsmRestartRecToFile(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

恢复因SsmPauseRecToFile函数而暂停的文件录音任务。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPauseRecToFile

2.10.3.10 SsmChkRecToFile

查询文件录音任务的完成情况。

函数原型：

int SsmChkRecToFile(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	文件录音任务正常结束
1	文件录音任务正在进行
2	文件录音任务被暂停

功能描述：

查询文件录音任务的完成情况。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecToFile，SsmRecToFileA，SsmRecToFileB，SsmRecToFileEx

2.10.3.11 SsmGetDataBytesToRecord

获取尚未完成的录音数据字节数。

函数原型：

int SsmGetDataBytesToRecord(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	尚未完成录制的语音数据字节数

功能描述：

获取文件录音任务启动后，尚未完成的录音数据字节数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRecTime

2.10.4 录制到指定内存的函数

2.10.4.1 SsmRecToMem

启动内存录音任务。

函数原型：

int SsmRecToMem(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, DWORD dwStartOffset)

参数说明：

ch	通道编号
nFormat	录音数据的编码格式。取值范围： 取值 / 编码格式 / 备注 6 / A-Law / 7 / μ-Law / 17 / IMA ADPCM / 要求板卡具有硬件编码能力 23 / VOX / 要求板卡具有硬件编码能力 49 / GSM / 85 / MP3 / 要求板卡具有硬件编码能力 131 / GC8 / 要求板卡具有硬件编码能力 65411 / G.729A / 要求板卡具有硬件编码能力 通道是否支持硬件编码器的详细内容请参见第 1 章中“SynCTI支持的CODEC”部分。
pBuf	录音缓冲区的首地址指针，由应用程序分配空间
dwBufSize	录音缓冲区的大小，单位为字节
dwStartOffset	起始录音位置在录音缓冲区中的偏移量。取值范围：0~dwBufSize-1。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

启动内存录音任务。

调用本函数后，驱动程序内部会启动一个内存录音任务，从参数 dwStartOffset 指定的偏移量开始将录音数据写入到 pBuf，同时维护一个初始值等于 dwStartOffset 的录音指针（偏移量），用于向提交的缓冲区中写入数据。当缓冲区写满后，自动从缓冲区头部开始循环写入。

当发生下列事件时，内存录音任务终止：

- ✧ 应用程序调用SsmStopRecToMem；
- ✧ 驱动程序检测到DTMF或对端挂机。详细内容请参见第 1 章中设置录音任务的终止条件部分。

函数SsmCheckRecord可以用来查询内存录音任务的完成情况。

在内存录音任务的执行期间，每当满足设定的条件时，驱动程序都会向应用程序抛出E_PROC_RecordMem事件。E_PROC_RecordMem事件的输出条件可以通过函数SsmSetEvent（携带E_PROC_RecordMem参数）进行设置，可以选择下列 3 种方式之一：

- ✧ 当驱动程序完成了指定时间长度的语音数据的录制；
- ✧ 驱动程序内部的录音指针越过缓冲区的 dwBufSize /2 位置；
- ✧ 驱动程序内部的放音指针到达缓冲末尾。

函数SsmStopRecToMem可以用来终止内存录音任务，函数SsmGetRecOffset可以用来获取驱动程序内部的录音指针。

注意事项：

- 有关选择录音信号源、设置音量的详细内容请参见第 1 章中设置录音信号源部分。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopRecToMem，SsmGetRecOffset，SsmSetDTMFStopRecCharSet，SsmSetHangupStopRecFlag，SsmSetRecVolume，SsmSetRecMixer，SsmSetRecBack

2.10.4.2 SsmStopRecToMem

终止内存录音任务。

函数原型：

int SsmStopRecToMem(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止内存录音任务。

注意事项：

- 录音操作停止后，驱动程序会抛出 E_PROC_RecordEnd 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecToMem

2.10.4.3 SsmGetRecOffset

获取内存录音任务执行过程中，驱动程序内部的录音指针。

函数原型：

int SsmGetRecOffset(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	驱动程序内部的录音指针，其值为相对于SsmRecToMem函数提交的缓冲区首地址的偏移量，单位为字节

功能描述：

获取内存录音任务执行过程中，驱动程序内部的录音指针。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecToMem

2.10.5 录制到指定乒乓缓存的函数（CTI 系列）

2.10.5.1 SsmRecordMemBlock

向驱动程序提交一个缓冲区，并启动乒乓内存录音任务。

函数原型：

int SsmRecordMemBlock(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, RECORDMEMBLOCKHANDLER pfnCallback, PVOID pV)

参数说明：

<table>
<tr><td>ch</td><td>通道编号</td></tr>
<tr><td>nFormat</td><td>录音数据的编码格式。取值范围：
<table>
<tr><th>取值</th><th>编码格式</th><th>备注</th></tr>
<tr><td>6</td><td>A-Law</td><td></td></tr>
<tr><td>7</td><td>μ-Law</td><td></td></tr>
<tr><td>17</td><td>IMA ADPCM</td><td>要求板卡具有硬件编码能力</td></tr>
<tr><td>23</td><td>VOX</td><td>要求板卡具有硬件编码能力</td></tr>
<tr><td>49</td><td>GSM</td><td></td></tr>
<tr><td>85</td><td>MP3</td><td>要求板卡具有硬件编码能力</td></tr>
<tr><td>131</td><td>GC8</td><td>要求板卡具有硬件编码能力</td></tr>
<tr><td>65411</td><td>G.729A</td><td>要求板卡具有硬件编码能力</td></tr>
</table>
通道是否支持硬件编码器的详细内容请参见第 1 章中“SynCTI支持的CODEC”部分。</td></tr>
<tr><td>pBuf</td><td>录音缓冲区指针</td></tr>
<tr><td>dwBufSize</td><td>录音缓冲区大小，单位为字节。dwBufSize 的最小值按照下面规则确定：
如果配置项RecordAndPlayUseAsIP设置为 0（缺省值），dwBufSize应不低于 768 字节；
如果RecordAndPlayUseAsIP设置为 1，对于A-率和μ-率，dwBufSize应不低于 192 字节（推荐值：Windows操作系统下应不低于 192 字节，Linux下应不低于 384 字节）；对于IMA ADPCM格式，缓冲区大小应不低于 768 字节</td></tr>
<tr><td>pfnCallback</td><td>回调函数指针，NULL 表示不设置回调函数。如果本参数不为 NULL，当驱动程序完成一个缓冲区的录制后会自动调用 pfnCallback 设置的回调函数。回调函数的原型在 shpa3api.h 中声明为：
typedef BOOL (* RECORDMEMBLOCKHANDLER)
(int ch, int nEndReason, PUCHAR pucBuf, DWORD dwStopOffset, PVOID pVoid);
其中，
ch：通道编号；
nEndReason：结束原因。取值范围：
1：录音任务被应用程序终止。
2：录音任务因检测到 DTMF 按键字符而结束。
3：录音任务因检测到对端挂机而结束。
4：缓冲区播放完毕。
pucBuf：录制完毕的缓冲区指针；
dwStopOffset：缓冲区录制完毕时，驱动程序内部的录音指针在缓冲区中偏移量，单位为字节。
pVoid：应用程序在调用双缓冲内存录音函数时传递的指针。</td></tr>
<tr><td>pVoid</td><td>PVOID 类型指针，通常用于应用程序向回调函数传递其数据结构。这个指针将由驱动程序透明地传递给回调函数</td></tr>
</table>

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

向驱动程序提交一个缓冲区，并启动乒乓缓冲区录音任务。有关乒乓缓冲区录音的详细内容请参见第 1 章中乒

乒缓冲区录音部分。

当发生下列事件时，乒乓缓冲区录音任务终止：

- ✧ 驱动程序完成全部已提交的缓冲区的录制；
- ✧ 应用程序调用SsmStopRecordMemBlock；
- ✧ 驱动程序检测到DTMF或对端挂机。详细内容请参见第 1 章中设置录音任务的终止条件部分。

函数SsmCheckRecord可以用来查询内存序列播放任务的完成情况。

注意事项：

- 有关选择录音信号源、设置音量的详细内容请参见第 1 章中设置录音信号源部分；
- 应用程序可以随时调用函数SsmStopRecordMemBlock终止乒乓缓冲区录音任务，该函数不能在回调函数中调用；
- 在回调函数中，应用程序的代码执行时间应尽量短，不能大于驱动程序的硬件中断时间（8 毫秒），否则可能引起驱动程序出错；
- 乒乓缓冲区录音任务终止后，驱动程序会向应用程序抛出 E_PROC_PlayEnd 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopRecordMemBlock，SsmRecToMem

2.10.5.2SsmStopRecordMemBlock

终止乒乓缓冲区录音任务。

函数原型：

int SsmStopRecordMemBlock (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

终止乒乓缓冲区录音任务。

注意事项：

- 本函数不能在SsmRecordMemBlock设置的回调函数中调用；
- 录音操作停止后，驱动程序会抛出 E_PROC_RecordEnd 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmRecordMemBlock

2.11 有关交换总线的函数

2.11.1 双向连接函数（CTI 系列）

2.11.1.1 建立双向连接

2.11.1.1.1SsmTalkWith

2.11.1.1.2SsmTalkWithEx

建立 2 个通道之间的双向连接。SsmTalkWithEx 可以设置参与交换的 2 个通道的音量。

函数原型：

int SsmTalkWith(int ch1,int ch2)

int SsmTalkWithEx(int ch1, int nVlm1, int ch2, int nVlm2)

参数说明：

ch1	通道编号 1
ch2	通道编号 2
nVlm1	ch1 的音量，取值范围：-6~+6，负值表示衰减，正值表示提升，缺省值为 0。取值乘以 3 即为分贝值。
nVlm2	ch2 的音量，取值范围：-6~+6，负值表示衰减，正值表示提升，缺省值为 0。取值乘以 3 即为分贝值。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

建立指定通道 ch1 到 ch2 的双向连接，即 ch1 可以听到 ch2 的声音，ch2 也可以听到 ch1 的声音。如果 ch1 或 ch2 上的来话信号还没有上行到交换总线，驱动程序会自动为其申请一个交换总线时隙，并自动将来话信号上行到该时隙。

注意事项：

- 本函数只适用于SHT/SHD/SHN系列板卡，参数nVlm1 和nVlm2 用来设置第 1 章“SHT系列板卡的工作原理框图”或“SHD系列板卡的工作原理框图” 或 “SHN系列板卡的工作原理框图” 中的音量调节器A5。A5 的增益也可以通过配置项DefaultSpeakVolume进行设置，缺省值为 0DB；
- 如果板卡的型号中不包含“-CT”字样，则 ch1 和 ch2 必须在同一片板卡上，否则函数会失败；
- 拆除连接时，应使用SsmStopTalkWith函数，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopTalkWith，SsmListenTo，SsmListenToEx，SsmLinkFrom，SsmLinkFromEx

2.11.1.2拆除双向连接

2.11.1.2.1SsmStopTalkWith

拆除 2 个通道之间的双向连接。

函数原型：

int SsmStopTalkWith(int ch1,int ch2)

参数说明：

ch1	听者/说者通道编号
ch2	说者/听者通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

拆除 ch1 到 ch2 的双向连接。

注意事项：

- 本函数只适用于 SHT/SHD/SHN 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmTalkWith，SsmTalkWithEx

2.11.2 单向连接函数

2.11.2.1 建立单向连接

2.11.2.1.1SsmListenTo

2.11.2.1.2SsmListenToEx

2.11.2.1.3SsmLinkFrom

2.11.2.1.4SsmLinkFromEx

2.11.2.1.5SsmLinkFromAllCh

建立说者通道到听者通道的单向连接。使用 SsmListenTo、SsmListenToEx 建立两个通道之间的单向连接，并

且一个通道只能听一个通道；使用 SsmLinkFrom、SsmLinkFromEx 建立的两个通道之间的单向连接，一个通道可以同时听多个通道。SsmListenToEx、SsmLinkFromEx 还可以在设置单向连接的音量。SsmLinkFromAllCh 可以一次建立单个说者通道到多个听者通道单向连接。

函数原型：

int SsmListenTo(int nSourceCh, int nListeningCh)
int SsmListenToEx(int nSourceCh, int nVolume, int nListeningCh)
int SsmLinkFrom(int nSourceCh, int nListeningCh)
int SsmLinkFromEx(int nSourceCh, int nVolume, int nListeningCh)
int SsmLinkFromAllCh(int nSourceCh, int nVolume, int* pnListenerTable,int nListenerNum)

参数说明：

nSourceCh	说者通道编号
nListeningCh	听者通道编号。 **注意：如果 nListeningCh 为 ATP 或 DST 系列板卡的通道 0，说者的声音将被送到板载扬声器端口。**
nVolume	说者的音量，取值范围：-6~+6，正值表示提升，负值表示衰减，缺省值为 0。取值乘以 3 即为分贝值。 nVolume用来设置来话信号进入总线前的音量调节器A5，详细内容请参见第 1 章中板卡的工作原理框图部分。A5 的增益也可以通过配置项DefaultSpeakVolume进行设置，缺省值为 0DB。
pnListenerTable	听者通道表指针，存储空间由应用程序分配。听者通道必须连续存放。
nListenerNum	pnListenerTable 中听者通道的数目

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

建立说者通道到听者通道的单向连接，即用 nListeningCh 听 nSourceCh 的声音，但 nSourceCh 无法听到 nListeningCh 的声音。

SsmListenToEx除了具备SsmListenTo的全部功能外，还能设置连接时的音量。使用SsmListenTo、SsmListenToEx建立的单向连接，一个通道在同一时刻只能听一个通道。使用SsmListenTo、SsmListenToEx建立的连接，应使用SsmStopListenTo函数来拆除连接，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

SsmLinkFromEx除了具备SsmLinkFrom的全部功能外，还能设置连接时的音量。使用SsmLinkFrom、SsmLinkFromEx建立的单向连接应使用SsmStopLinkFrom函数来拆除连接，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

SsmLinkFromAllCh可以一次建立单个说者通道到多个听者通道单向连接，相当于对多个听者通道连续调用SsmLinkFromEx。使用SsmLinkFromAllCh建立的单向连接应使用SsmUnLinkFromAllCh函数来拆除连接，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

SsmLinkFrom、SsmLinkFromEx、SsmLinkFromAllCh 允许一个通道同时听多个说者通道的声音，这些说者通道的声音通过 TDM 总线交换到听者通道的会议混音器中，经过混音后的声音作为去话送给听者。

注意事项：

- 如果板卡的型号中不包含“-CT”字样，nSourceCh 和 nListeningCh 必须在同一片板卡上，否则函数会失败；
- 对于SHT/SHD/SHN系列板卡，由于下总线混音器M2 具有 6 个输入信号源A_{4-1}~A_{4-6}，因此，每个通道最多可以同时听 6 个通道的声音，即，一个通道最多可以调用 6 次SsmLinkFrom、SsmLinkFromEx函数。详细内容请参见第 1 章中“SHT系列板卡的工作原理框图”部分或“SHD系列板卡的工作原理框图”部分或“SHN系列板卡的工作原理框图”部分。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SmStopListenTo，SsmStopLinkFrom，SsmUnLinkFromAllCh，SsmTalkWith，SsmTalkWithEx

代码范例：

本示例代码实现下列功能：

- ✓ 通道 0 听通道 5 的声音；
- ✓ 通道 1 听通道 6 的声音（音量提升 6 个单位）；
- ✓ 通道 2 想要同时听通道 9、通道 10、通道 11 的声音；
- ✓ 通道 3 想要同时听通道 9（正常音量）、通道 10（音量衰减 1 个单位）、通道 11（音量提升 2 个单位）；
- ✓ 通道 4 的声音要同时送给通道 12、13、14 听（缺省音量）

代码实现：

```
SsmListenTo（ 5，0）;                    // 为听者通道 0 建立连接

SsmListenToEx（ 6，6，1）;               // 为听者通道 1 建立连接

SsmLinkFrom（ 9，2）;                    // 为听者通道 2 建立连接
SsmLinkFrom（10，2）;
SsmLinkFrom（11，2）;

SsmLinkFromEx（ 9，  0，3）;             // 为听者通道 3 建立连接
SsmLinkFromEx（10，-1，3）;
SsmLinkFromEx（11，  2，3）;

int ListenerChList[3]={12，13，14};      // 建立说者通道 4 到听者通道 12、13、14 的单向连接
SsmLinkFromAllCh（4，0，ListenerChList，3）;

    ......                               // 应用程序的其它操作

SmStopListenTo（ 5，0）;                 // 为听者通道 0 拆除连接

SmStopListenTo（ 6，1）;                 // 为听者通道 0 拆除连接

SsmStopLinkFrom（ 9，2）;                // 为听者通道 1 拆除连接
SsmStopLinkFrom（10，2）;
SsmStopLinkFrom（11，2）;

SsmStopLinkFrom（ 9，3）;                // 为听者通道 2 拆除连接
SsmStopLinkFrom（10，3）;
SsmStopLinkFrom（11，3）;

SsmLinkFromAllCh（4，ListenerChList，3）; // 拆除通道 4 到听者通道 12、13、14 的单向连接
```

2.11.2.2 拆除单向连接

2.11.2.2.1 SsmStopListenTo

2.11.2.2.2 SsmStopLinkFrom

2.11.2.2.3 SsmUnLinkFromAllCh

拆除说者通道到听者通道的单向连接。SsmStopListenTo 用于拆除 SsmListenTo 或 SsmListenToEx 建立的单向连接，SsmStopLinkFrom 用于拆除 SsmLinkFrom 或 SsmLinkFromEx 建立的单向连接，SsmUnLinkFromAllCh 用于拆除 SsmLinkFromAllCh 建立的单向连接。

函数原型：

int SsmStopListenTo(int nSourceCh,int nListeningCh)
int SsmStopLinkFrom(int nSourceCh,int nListeningCh)
int SsmUnLinkFromAllCh (int nSourceCh, int* pnListenerTable,int nListenerNum)

参数说明：

nSourceCh	说者通道编号
nListeningCh	听者通道编号
pnListenerTable	听者通道表指针，存储空间由应用程序分配。听者通道必须连续存放。
nListenerNum	pnListenerTable 中听者通道的数目

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

拆除说者通道到听者通道的单向连接。

注意事项：

- 建立单向连接的函数和拆除单向连接的函数必须配对使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmListenTo，SsmListenToEx，SsmLinkFrom，SsmLinkFromEx，SsmLinkFromAllCh

2.11.3 获取总线信息函数

2.11.3.1 SsmGetChBusInfo

获取与 ch 通道相关的总线信息。

函数原型：

int WINAPI SsmGetChBusInfo(int ch, PBUS_OP* p)

参数说明：

ch	通道编号
p	Bus 结构指针

返回值：

-1	通道没有总线连接
>=0	与 ch 通道连接的通道号

功能描述：

取得与 ch 通道相关的总线参数。

Bus 结构说明

```
{       BOOL    bEnHwOpBus;                 总线开关
        BOOL    bEnHwOpSetLinkFromVlm;      总线连接音量设置开关
        int     nST;                        总线默认连接的码流
        int     nTs;                        总线默认连接的时隙
        int     nToBusCh;                   默认连接所占用的总时隙
        int     nPlayST;                    录音模块放音上总线默认连接的码流
        int     nPlayTs;                    录音模块放音上总线默认连接的时隙
        int     nPlayToBusCh;               录音模块放音上总线所占用的总时隙
        int     nSpeakerVlm;                音量
        int     nTotListener;               电话会议总听众数
        int *   pnListenerCh; /             电话会议总成员数
        int     nFromSpeaker;               发言人数
        int     nDefaultSpeakerVlm; }       总线连接音量
```

该 Bus 结构是驱动内部定义结构，有些对应用层是没有意义的，用户可不必关心。

注意事项：

- 该函数的声明在 STBUS.h。

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.11.4 高级总线操作函数（CTI 系列）

2.11.4.1 强制拆除通道的所有总线连接

2.11.4.1.1 SsmClearChBusLink

拆除通道上建立的所有总线连接，使通道恢复到驱动程序初始化完成时的总线连接状态。

函数原型：

int SsmClearChBusLink(int nCh)

参数说明：

nCh	监听通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

拆除在通道 nCh 上调用任何函数所建立的总线连接，包括：

- SsmListenTo、SsmListenToEx、SsmLinkFrom、SsmLinkFromEx、SsmLinkFromAllCh
- SsmTalkWith、SsmTalkWithEx
- SsmLinkFromBus、SsmLinkFromBusEx

使通道恢复到驱动程序初始化完成时的总线连接状态。

注意事项：

- 如果通道已经加入了一个电话会议室中，其总线连接也将被强制拆除。

相关信息：

驱动程序版本要求	SynCTI Ver.4.5.6.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.11.4.2 通道到总线时隙的单向连接

2.11.4.2.1SsmLinkToBus

改变上总线混音器的输出信号上行到总线的时隙。

函数原型：

int SsmLinkToBus(int ch, int ts)

参数说明：

ch	通道编号
ts	可用的TDM总线公共时隙编号，取值范围：N~4095，N为系统中的通道总数的 2 倍。有关可用的公共时隙编号请参见第 1 章中“TDM交换总线”部分。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

板卡初始化时已经将通道ch的上总线混音器的输出信号连接到了某个公共时隙（假设为ts0），本函数将该输出信号重新连接到时隙ts。拆除连接应使用SsmUnLinkToBus函数。

音量调节器A5 用于设置来话信号进入总线前的音量，详细内容请参见第 1 章中板卡的工作原理框图部分。A5 的增益也可以通过配置项DefaultSpeakVolume进行设置，缺省值为 0DB。

注意事项：

- 时隙 ts 必须是未被使用的公共时隙；

- 本函数必须与SsmUnLinkToBus成对调用，否则会引起信号混乱；
- 本函数是底层函数，只适用于将 Synway 板卡与第 3 方板卡进行互联的场合。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnLinkToBus, SsmLinkFromBus,SsmLinkFromBusEx

2.11.4.2.2SsmUnLinkToBus

恢复将上总线混音器的输出信号连接到缺省的总线时隙。

函数原型：

int SsmUnLinkToBus(int ch, int ts)

参数说明：

ch	通道编号
ts	时隙编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

拆除通道 ch 到总线时隙 ts 的单向连接，并恢复将上总线混音器的输出信号连接到缺省的总线时隙。

注意事项：

- 本函数必须与SsmLinkToBus成对调用，否则会引起信号混乱；
- 本函数是底层函数，只适用于将 Synway 板卡与第 3 方板卡进行互联的场合。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmLinkToBus

2.11.4.3 总线时隙到通道的单向连接

2.11.4.3.1SsmLinkFromBus

2.11.4.3.2SsmLinkFromBusEx

建立总线时隙到听者通道的单向连接。SsmLinkFromBusEx 除了具备 SsmLinkFromBus 的全部功能外，还可以设置音量。

函数原型：

int SsmLinkFromBus(int ts, int ch)

int SsmLinkFromBusEx(int ts, int ch, int nVolume)

参数说明：

ch	通道编号
ts	时隙编号
nVolume	ts 时隙上的数据进入通道 ch 时的音量，取值范围：-6~+6，正值表示提升，负值表示衰减，缺省值为 0。取值乘以 3 即为分贝值。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

建立总线时隙 ts 到听者通道 ch 的单向连接。

注意事项：

- SsmLinkFromBus或SsmLinkFromBusEx必须与SsmUnLinkFromBus成对调用，否则会引起信号混乱。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmLinkToBus, SsmUnLinkFromBus

2.11.4.3.3 SsmUnLinkFromBus

拆除总线时隙到听者通道的单向连接。

函数原型：

int SsmUnLinkFromBus(int ts, int ch)

参数说明：

ch	通道编号
ts	时隙编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

拆除总线时隙 ts 到听者通道 ch 的单向连接。

注意事项：

- SsmLinkFromBus或SsmLinkFromBusEx必须与SsmUnLinkFromBus成对调用，否则会引起信号混乱。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmLinkFromBus，SsmLinkFromBusEx

2.11.4.3.4SsmListenToPlay

实现通道对录音通道的放音监听。

函数原型：

int WINAPI SsmListenToPlay(int ch1, int vlm, int ch2)

参数说明：

ch1	监听通道编号
ch2	放音通道编号，要求该通道类型为录音通道
vlm	放音的音量

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

实现通道 ch1 对录音通道 ch2 的放音监听。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnListenToPlay

2.11.4.3.5SsmUnListenToPlay

结束通道对录音通话的放音监听。

函数原型：

int WINAPI SsmUnListenToPlay(int ch1, int ch2)

参数说明：

ch1	监听通道编号
ch2	放音通道编号，要求该通道类型为录音通道。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

结束通道 ch1 对录音通道 ch2 的放音监听。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmListenToPlay

2.12 有关板载音频功率放大器的函数

2.12.1 SsmQueryOpPowerAmp

查询通道是否具有板载音频放大电路。

函数原型：

int SsmQueryOpPowerAmp(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	本通道没有板载音频放大电路
1	本通道具有板载音频放大电路。

功能描述：

查询通道是否具有板载音频放大电路。有关板载音频放大电路的内容请参见对应板卡的原理框图部分。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetPowerAmpVlm，SetVolume

2.12.2 SsmSetPowerAmpVlm

2.12.3 SetVolume

设置板载功率放大器的增益。SetVolume 是较早版本的函数，推荐使用 SsmSetPowerAmpVlm。

函数原型：

void SsmSetPowerAmpVlm(int ch, int nGain)

void SetVolume(DWORD dwBoardId, DWORD nGain)

参数说明：

ch	通道编号。**注意：ch 对应的板卡内部编号必须为 0，否则函数会失败。**
dwBoardId	板卡 ID 编号
nGain	功率放大器的增益，取值范围：0~7，其中： 7： 0DB 6~1： -6×nGain DB

	0： 完全关闭 缺省值为 3（-18DB）

返回值：无

功能描述：

设置板载功率放大器的增益。有关板载音频放大电路的内容请参见对应板卡的原理框图部分。

注意事项：

- 只有 ATP 系列、SHT 系列或 DST 系列板卡的 0 通道上才具有板载功率放大器。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmQueryOpPowerAmp

2.13 有关板载扬声器的函数

2.13.1 SsmSetLine0OutTo

关闭或开启 USB 录音盒/语音盒中的内置板载扬声器。

函数原型：

int WINAPI SsmSetLine0OutTo(BOOL bEnable)

参数说明：

bEnable	=FALSE：将声音送到板载扬声器进行播放 =TRUE： 将声音送到板载扬声器插座，由外置扬声器进行播放

返回值：

-1	调用失败
0	操作成功

功能描述：

开启或关闭USB录音盒/语音盒中的内置板载扬声器。更多信息请参见第 1 章中“ATP系列板卡的工作原理图”或“SHT系列板卡的工作原理图”部分内容。

注意事项：

- 本函数只适用于 ATP 系列的 USB 录音盒、SHT 系列的 USB 语音盒；
- 配置项USBLine0Output可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.7 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetPowerAmpVlm

2.14 有关回波抵消器（Echo Canceller）的函数（CTI 系列）

2.14.1 设置回波抵消器的工作状态

2.14.1.1 SsmQueryOpEchoCanceller

查询通道是否支持回波抵消。

函数原型：

int SsmQueryOpEchoCanceller(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	不支持
1	支持

功能描述：

查询通道是否支持回波抵消。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.14.1.2 SsmSetEchoCanceller

设置回波抵消器的工作状态。

函数原型：

int SsmSetEchoCanceller(int ch, BOOL bEnable)

参数说明：

ch	通道编号
bEnable	TRUE： 开启 FALSE：关闭

返回值：

-1	调用失败
0	调用成功

功能描述：

设置回波抵消器的工作状态。

注意事项：

- 配置项EnableEchoCancellor可以实现相同功能，缺省值为开启；
- 详细内容请参见第 1 章中回波抵消器部分；
- 本函数只支持 SHD、SHT 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetEchoCancellerState

2.14.1.3 SsmGetEchoCancellerState

获取回波抵消器的工作状态。

函数原型：

int SsmGetEchoCancellerState(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	关闭
1	开启

功能描述：

获取回波抵消器的工作状态。

注意事项：

- 本函数只支持 SHD、SHT 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetEchoCanceller

2.14.2 设置回波抵消器的工作参数

2.14.2.1 SsmSetEchoCancelDelaySize

设置回波抵消器的滤波延迟系数。

函数原型：

int SsmSetEchoCancelDelaySize(int ch,WORD wSize)

参数说明：

ch	通道编号
wSize	滤波延迟系数，取值范围：0~31

返回值：

-1	调用失败
0	成功

功能描述：

设置回波抵消器的滤波延迟系数。

注意事项：

- 本函数只支持 SHD、SHT 系列板卡；
- 配置项EchoCancelDelaySize可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetEchoCancelDelaySize

2.14.2.2 SsmGetEchoCancelDelaySize

获取回波抵消器的滤波延迟系数。

函数原型：

int SsmGetEchoCancelDelaySize(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	滤波延迟系数

功能描述：

获取指定通道上的滤波延迟系数。

注意事项：

- 本函数只支持 SHD、SHT 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetEchoCancelDelaySize

2.15 有关电话会议的函数（CTI 系列）

2.15.1 会议室初始化函数

2.15.1.1 SsmCreateConfGroup

创建一个会议室。

函数原型：

int SsmCreateConfGroup(int nMaxMember, int nMaxSpeaker, int nMaxSpeaking, int nMaxSilenceTime)

参数说明：

nMaxMember	会议室中的最大通道数，取值范围： -1： 使用配置项ConfDefaultMaxGroupMember指定的参数 3~N：N 为应用系统中的通道总数
nMaxSpeaker	设置具有发言权的会议通道的最大个数。主持人、主席、动态发言模式均具有发言权，背景音乐模式是否具有发言权取决于配置项BackgroundVoicePriority的设置。取值范围：

	-1：使用配置项ConfDefaultMaxGroupSpeaker指定的参数 ≥1：会议通道总数
nMaxSpeaking	允许同时发言的最大成员数，取值范围： -1： 使用配置项ConfDefaultMaxGroupSpeaking指定的参数 1~6： 设定值
nMaxSilenceTime	静默的最小保持时间。取值范围： -1：使用配置项ConfDefaultMaxSilenceTime指定的参数 ≥1：设定值，单位为秒 更多信息请参见第 1 章中“分布式电话会议”部分内容。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
≥0	新创建的电话会议室的 ID 号

功能描述：

创建一个会议室。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFreeConfGroup，SsmJoinConfGroup，SsmGetConfGrpInfo，SsmGetConfGrpCfgInfo

2.15.1.2 SsmFreeConfGroup

注销一个会议室。

函数原型：

void SsmFreeConfGroup(int nGrpId)

参数说明：

nGrpId	会议室编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

注销一个会议室，释放相关的资源。

注意事项：

- 如果会议室中还有尚未退出会议的通道，驱动程序会自动调用SsmExitConfGroup函数将其退出会议室。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmCreateConfGroup

2.15.2 会议室成员操作函数

2.15.2.1 SsmJoinConfGroup

将通道加入到会议室。

函数原型：

int SsmJoinConfGroup(int nGrpId, int ch, WORD wJoinMode, int nMixerVolume, BOOL bCreateAlways, BOOL bExitGrpAlways)

参数说明：

nGrpId	会议室编号
ch	通道编号
wJoinMode	通道的发言模式，取值范围： 0：主持人模式； 1：动态发言模式； 2：听众模式； 3：背景音乐模式，由配置项 PlayVoiceIsListen 决定能否听取其它通道的发言：当 PlayVoiceIsListen＝0 时，只有发言权，无法听取其他通道的声音，用于对会议放音； 当 PlayVoiceIsListen＝1 时，既有发言权，也能听取其他通道的声音（驱动默认方式）。 4：主席模式 5：动态只发言模式 有关发言模式及会议调度算法的更多信息请参见第 1 章中“分布式电话会议”部分内容。
nMixerVolume	本通道来话信号进入会议混音器前的音量，取值范围：-6~+6，大于 0 表示音量提升，小于 0 表示音量衰减。取值乘以 3 即为分贝值。
bCreateAlways	调用本函数时，如果参数 nGrpId 指定的会议室尚未创建，本参数指示是否由驱动程序以 nGrpId 作为会议室编号自动创建一个新的会议室。 TRUE： 是 FALSE：否，函数将返回失败。
bExitGrpAlways	如果通道已经加入到其它的会议室，本参数指示是否由驱动程序自动将通道退出该会议室。 TRUE： 是 FALSE：否，函数将返回失败。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

将通道加入到会议室。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmExitConfGroup，SsmSetListenVlmInConf

2.15.2.2 SsmExitConfGroup

将通道退出会议室。

函数原型：

int SsmExitConfGroup(int ch, BOOL bFreeGrpAlways)

参数说明：

ch	通道编号
bFreeGrpAlways	如果会议室中只剩下本通道，本参数指示是否由驱动程序自动注销会议室。 TRUE：注销； FALSE：不注销

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

将通道退出会议室。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmJoinConfGroup

2.15.2.3SsmSetListenVlmInConf

设置会议通道上的来话信号在下总线时的音量。

函数原型：

int SsmSetListenVlmInConf(int ch, in nVlm)

参数说明：

ch	通道号
nVlm	下总线音量

返回值：

0	函数调用成功
-1	函数调用失败

功能描述：

设置会议通道上的来话信号在下总线时的音量，即设置 SHT/SHD/SHN 系列板卡原理框图中的音量调节器 A5。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmJoinConfGroup

2.15.3 获取会议室信息的函数

2.15.3.1 SsmGetConfCfgInfo

获取分布式电话会议的配置参数。

函数原型：

int SsmGetConfCfgInfo(PWORD pwMaxMember, PWORD pwMaxSpeaker, PWORD pwMaxSpeaking, PWORD pwMaxSilenceTime)

参数说明：

pwMaxMember	返回配置项ConfDefaultMaxGroupMember的设定值
pwMaxSpeaker	返回配置项ConfDefaultMaxGroupSpeaker的设定值
pwMaxSpeaking	返回配置项ConfDefaultMaxGroupSpeaking的设定值
pwMaxSilenceTime	返回配置项ConfDefaultMaxSilenceTime的设定值

返回值：

-1	调用失败
≥0	返回配置项ConfMaxGroup的设定值

功能描述：

获取分布式电话会议的配置参数。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfGrpInfo

2.15.3.2 SsmGetTotalConfGroup

获取会议室的实际数量。

函数原型：

int SsmGetTotalConfGroup()

参数说明：无

返回值：

-1	调用失败
≥0	会议室的实际数量

功能描述：

获取会议室的实际数量。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfGrpId

2.15.3.3 SsmGetConfGrpCfgInfo

获取会议室的工作参数的设定值。

函数原型：

int SsmGetConfGrpCfgInfo(int nGrpId, PWORD pwMaxMember, PWORD pwMaxSpeaker, PWORD pwMaxSpeaking, PWORD pwMaxSilenceTime)

参数说明：

nGrpId	会议室编号
pwMaxMember	返回会议通道的总数
pwMaxSpeaker	返回具有发言权的会议通道的总数

pwMaxSpeaking	返回允许同时发言的会议通道的总数
pwMaxSilenceTime	返回会议通道的最小静默保持时间，单位为秒

返回值：

-1	调用失败
0	调用成功

功能描述：

获取会议室工作参数的设定值，即应用程序调用SsmCreateConfGroup函数创建会议室时设定的参数。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfCfgInfo

2.15.3.4 SsmGetConfGrpInfo

获取会议室的统计信息。

函数原型：

int SsmGetConfGrpInfo(int nGrpId, PWORD pwTotalMember, PWORD pwTotalSpeaker, PWORD pwTotalSpeaking)

参数说明：

nGrpId	会议室编号
pwTotalMember	返回会议通道的实际数量
pwTotalSpeaker	返回会议室中具有发言权通道的实际数量
pwTotalSpeaking	正在使用会议混音器的会议通道的实际数量

返回值：

-1	调用失败
0	调用成功

功能描述：

获取会议室的统计信息。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfCfgInfo

2.15.3.5 SsmGetConfGrpId

获取全部会议室的编号表。

函数原型：

int SsmGetConfGrpId(int* pnGrpId)

参数说明：

pnGrpId	返回会议室编号的数组指针

返回值：

-1	调用失败
0	调用成功

功能描述：

获取全部会议室的编号表。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmValidateGrpId

2.15.3.6 SsmValidateGrpId

查询会议室是否已经被创建。

函数原型：

int SsmValidateGrpId(int nGrpId)

参数说明：

nGrpId	会议室编号

返回值：

1	会议室已经创建
0	会议室尚未创建

功能描述：

查询会议室是否已经被创建。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfGrpId

2.15.4 获取会议室成员信息的函数

2.15.4.1 SsmGetConfGrpMmbrId

获取已经加入会议室的会议通道编号。

函数原型：

int SsmGetConfGrpMmbrId(int nGrpId, int* pnMmbrId)

参数说明：

nGrpId	会议室编号
pnMmbrId	返回会议通道编号的数组指针

返回值：

-1	调用失败
0	调用成功

功能描述：

获取已经加入会议室的会议通道编号。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfGrpMmbrInfo

2.15.4.2SsmGetConfGrpMmbrInfo

获取会议室中的会议通道的详细信息。

函数原型：

int SsmGetConfGrpMmbrInfo(int nGrpId, int nMmbrId, int* pnAppCh, PWORD pwJoinMode, PWORD pwIsSpeaking, PDWORD pdwSilenceTime)

参数说明：

nGrpId	会议室编号
nMmbrId	会议通道的编号
pnAppCh	返回会议通道编号对应的通道逻辑编号
pwJoinMode	返回会议通道发言模式
pwIsSpeaking	返回会议通道是否正在发言
pdwSilenceTime	返回会议通道保持静默的时间，单位为秒

返回值：

-1	调用失败
0	调用成功

功能描述：

根据会议室编号和会议通道编号，获取会议室中的会议通道的详细信息。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfChInfo

2.15.4.3 SsmGetConfChInfo

查询通道是否加入了某个会议室。

函数原型：

int SsmGetConfChInfo(int ch, int* pnGrpId, int* pnMmbrId, PWORD pwJoinMode, PWORD pwIsSpeaking, PDWORD pdwSilenceTime)

参数说明：

ch	通道编号
pnGrpId	返回会议室的编号
pnMmbrId	返回 ch 在会议室中的成员编号
pwJoinMode	返回 ch 加入会议室的发言模式

pwIsSpeaking	返回 ch 是否正在发言的标志
pdwSilenceTime	返回 ch 保持静默的时间，单位为秒

返回值：

-1	调用失败
0	调用成功

功能描述：

查询通道是否加入了某个会议室。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetConfGrpMmbrInfo

2.16 有关 FSK 收发器的函数（CTI 系列）

2.16.1 FSK 数据发送器

2.16.1.1 SsmSetFskPara

设置 FSK 数据发送器的工作参数。

函数原型：

int SsmSetFskPara(int nFreq0, int nFreq1, int nBaudrate, int nMdlAmp)

参数说明：

nFreq0	比特 0 的载波频率，单位：Hz，取值范围：300~3400。本参数也可以通过配置项FreqBit0进行设置，缺省值未 2200Hz
nFreq1	比特 0 的载波频率，单位：Hz，取值范围：300~3400。本参数也可以通过配置项FreqBit1进行设置，缺省值为 1200Hz
nBaudrate	波特率，单位：bps，取值范围：300~2400。本参数也可以通过配置项Baudrate进行设置，缺省值为 1200bps
nMdlAmp	调制幅度，取值范围：0~255。本参数也可以通过配置项MdlAmp进行设置，缺省值为 128

返回值：

-1	调用失败
0	调用成功

功能描述：

设置 FSK 数据发送器的工作参数。

注意事项：

- 如果接收端也使用 Synway 板卡，通常无需设置，使用相关配置项的缺省值即可。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetFskPara

2.16.1.2 SsmGetFskPara

获取 FSK 数据发送器的工作参数。

函数原型：

int SsmGetFskPara(int *nFreqBit0, int *nFreqBit1, int *nBaudrate, int *nMdlAmp)

参数说明：

nFreqBit0	返回比特 0 的载波频率，单位为 Hz
nFreqBit1	返回比特 1 的载波频率，单位为 Hz
nBaudrate	返回二进制码流的波特率，单位为 bps
nMdlAmp	返回调制信号的幅度

返回值：

-1	调用失败
0	调用成功

功能描述：

获取 FSK 数据发送器的工作参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetFskPara

2.16.1.3 SsmTransFskData

将数据转换为 BFSK 格式的二进制数据流。

函数原型：

int SsmTransFskData(unsigned char* pSrc, int nSrcLen,int nSyncLen,int nSyncOffLen,unsigned char* pStream)

参数说明：

pSrc	存放数据的缓冲区指针
nSrcLen	pSrc 中数据的长度，单位为字节
nSyncLen	同步码的比特个数，通常为 8 的整数倍，可以根据实际需要确定，建议设置为 96。
nSyncOffLen	标志码的比特个数，通常为 8 的整数倍，可以根据实际需要确定，建议设置为 32。
pStream	存放转换后的 BFSK 数据流的缓冲区指针，由应用程序分配空间，大小可以按照下面公式计算： (nSyncLen + nSyncOffLen + nSrcLen×10) / 8 + 1

返回值：

-1	调用失败
≥0	pStream 中实际的比特个数

功能描述：

将数据转换为BFSK格式的二进制数据流。有关BFSK数据流格式的详细内容请参见第 1 章中“FSK数据收发器”部分。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartSendFSK

2.16.1.4 SsmStartSendFSK

启动 FSK 数据发送器。

函数原型：

int SsmStartSendFSK(int ch, LPSTR pStream, DWORD dwMaxBit)

参数说明：

ch	通道编号
pStream	存放BFSK数据流的数据缓冲区指针。更多信息请参见第 1 章中“FSK数据收发器”部分内容
dwMaxBit	pStream 中有效的比特总数，可以是任意长度。

返回值：

-1	调用失败
0	调用成功

功能描述：

启动 FSK 数据发送器。

FSK数据发送器的工作参数可以通过函数SsmSetFskPara或配置项FreqBit0、FreqBit1、Baudrate、MdlAmp进行设置。

FSK数据发送器启动后，应用程序可以随时调用函数SsmStopSendFsk终止。当驱动程序完成pStream中全部比特的发送后，会向应用程序抛出E_PROC_SendFSK事件。应用程序也可以调用函数SsmCheckSendFsk来查询FSK数据发送器的进展情况。

注意事项：

- 在 FSK 数据发送器完成全部数据的发送前，不能启动语音播放操作，不能使用信号音发生器和 DTMF 发生器；
- 在启动 FSK 数据发送器前，为保证数据的完整性，如果开启了回波消除功能，请将其关闭；
- 原始数据必须经函数SsmTransFskData转换为BFSK数据流后，才能调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmCheckSendFsk，SsmStopSendFsk，SsmTransFskData

代码范例：

```
char Data[]=”12345”;                    //发送字符串”12345”
unsigned char Stream[256];              //存放转换后的 BFSK 比特流
int nMaxBit = SsmTransFskData( (unsigned char*)Data, 5,96, 32, Stream);
SsmStartSendFSK(ch, (PCHAR)Stream, nMaxBit);
```

2.16.1.5 SsmCheckSendFsk

查询 FSK 数据发送器的进展情况。

函数原型：

int SsmCheckSendFsk(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	完成
1	未完成

功能描述：

查询 FSK 数据发送器的进展情况。

注意事项：

- 如果应用程序使用基于事件的编程方式，建议使用 E_PROC_SendFSK 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartSendFSK

2.16.1.6 SsmStopSendFsk

停止 FSK 数据发送器。

函数原型：

int SsmStopSendFsk(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

停止 FSK 数据发送器。

注意事项：

- 驱动程序执行完本函数后，会向应用程序抛出 E_PROC_SendFSK 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartSendFSK

2.16.2 FSK 数据接收器

2.16.2.1 SsmStartRcvFSK

启动 FSK 数据接收器。

函数原型：

int WINAPI SsmStartRcvFSK(int ch, WORD wTimeOut, WORD wMaxLen, UCHAR ucEndCode, WORD wEndCodeCount)

参数说明：

ch	通道编号
wTimeOut	定时器的时长，单位为毫秒。
wMaxLen	FSK 数据的最大长度，单位为字节，取值范围：1~260。
ucEndCode	用于指示 FSK 数据包结束的特征字符
wEndCodeCount	ucEndCode 连续出现的最少个数，取值范围：0~10。

返回值：

-1	调用失败，失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

启动 FSK 数据接收器，开始执行 FSK 数据接收任务。

FSK 数据接收器启动后，驱动程序会启动一个定时器，并按照下列逻辑决定是否终止本次接收任务：

- ✧ 如果在 FSK 数据中发现由参数 ucEndCode 设定的特征结束字符，并且该字符连续出现的次数大于或等于 wEndCodeCount，本次接收任务终止，驱动程序会向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 被设置为 2）；
- ✧ 如果 FSK 数据的长度（字节总数）大于或等于 wMaxLen，本次 FSK 数据接收任务终止，驱动程序向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 为 3）。
- ✧ 如果线路上的 BFSK 的信号消失，并且消失时间超过 100 毫秒，本次接收任务结束，驱动程序向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 为 1）。
- ✧ 如果定时器溢出，本次接收任务结束，驱动向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 为 1）。

注意事项：

- Synway 板卡的 FSK 数据接收器的波特率 1200bps，比特 0 的载波频率为 2200Hz，比特 1 的载波频率为 1200Hz，均不可通过函数或配置项进行修改。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRcvFSK， SsmStopRcvFSK，SsmStartRcvFSK_II

代码范例：

```
.....
SsmStartRcvFSK (0, 10000, 260, "#", 1);                    //启动 FSK 数据接收器

MESSAGE_INFO Event;
If ( SsmWaitForEvent (64, &Event) == 0 )                   //等待事件发生
{
    switch(Event.EventCode)
    {
        case E_PROC_RcvFSK:                                //FSK 数据接收器终止
            if(Event.dwParam == 2)                         //收齐带结束符'#'的 FSK 数据
            {
                unsigned char DataBuf[260];
                SsmGetRcvFSK(Event.nReference, DataBuf);//取出数据
                .....                                      //处理代码
            }
```

```
            ......
        break;
        ......
    }
}
```

2.16.2.2SsmStartRcvFSK_II

启动 FSK 接收任务。

函数原型：

int WINAPI SsmStartRcvFSK_II(int ch, WORD wTimeOut, WORD wMaxLen, PUCHAR pucMarkCodeBuf, UCHAR ucMarkCodeCount)

参数说明：

ch	通道编号
wTimeOut	定时器的时长，单位为毫秒。
wMaxLen	FSK 数据的最大长度，单位为字节，取值范围：1~260。
pucMarkCodeBuf	存放标志码的缓冲区，最多可以设置 10 个不同的标志码。
ucMarkCodeCount	pucMarkCodeBuf 中标志码的个数，取值范围：0~10。如果本参数设置为 0，参数 pucMarkCodeBuf 将被忽略。

返回值：

-1	调用失败，失败原因可以通过调用函数SsmGetLastErrMsg获得
0	启动 FSK 接收器成功

功能描述：

启动 FSK 数据接收器，开始执行 FSK 数据接收任务。

本函数支持 2 种接收方式：长度方式和帧方式。帧方式是指需要传输的 FSK 数据具有固定的帧结构，驱动程序需要根据帧的标志（即标志码）对帧进行定位和分离；长度方式是指驱动程序只检查接收数据的长度，当收到的 FSK 数据长度大于 wMaxLen 后，接收任务就结束。如果 ucMarkCodeCount 大于 0，驱动程序优先使用帧方式进行接收，否则，按照长度方式进行接收，此时，参数 wMaxLen 有效。

1 帧数据由 4 个字段组成：标志码字段、长度字段、数据体字段和校验字节字段，如下图所示：

标志码	数据长度	数据体	校验字节

标志码字段：　表示帧的开始，占用 1 个字节，用于帧的定位。标志码的具体取值可以由应用程序自行决定；
长度字段：　表示数据体部分的长度，单位为字节，占用 1 个字节；
数据体：　需要传输的数据，最长不能超过 255 个字节；
校验字节：　占用 1 个字节。注意：校验字节由应用程序执行产生，驱动程序并不检查数据的完整性。

FSK 数据接收器的执行逻辑为：

- ✧ 启动一个定时器，等待 FSK 数据到达；
- ✧ 标志码方式： FSK 数据到达后，驱动程序在接收数据的前 10 个字节中寻找由参数 pucMarkCodeBuf 设定的标志码。如果找到，表明驱动程序顺利完成帧的定位，当驱动程序在完成 1 帧数据的接收后，本次接收任务终止，驱动程序向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 被设置为 4）；如果无法找到标志码，转而按照长度方式进行接收；
- ✧ 长度方式：当 FSK 数据的长度超过参数 wMaxLen 的设定值后，本次接收任务终止，驱动程序向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 被设置为 3）；
- ✧ 如果线路上的 BFSK 的信号消失，并且消失时间超过 100 毫秒，本次接收任务终止，驱动程序向应用程序

抛出 E_PROC_RcvFSK 事件（如果已经收到的 fsk 数据长度小于 wMaxLen，dwParam 为 1，否则为 3）；

✧ 如果定时器溢出，本次接收任务终止，驱动程序向应用程序抛出 E_PROC_RcvFSK 事件（dwParam 为 1）。

注意事项：

- Synway 板卡的 FSK 数据接收器的波特率 1200bps，比特 0 的载波频率为 2200Hz，比特 1 的载波频率为 1200Hz，均不可通过函数或配置项进行修改。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRcvFSK，SsmStopRcvFSK，SsmStartRcvFSK

代码范例：

```
.....
SsmStartRcvFSK_II (0, 10000, 1, "#", 1);                    //启动 FSK 数据接收器，

MESSAGE_INFO Event;
If ( SsmWaitForEvent (64, &Event) == 0 )                    //等待事件发生
{
    switch(Event.EventCode)
    {
        case E_PROC_RcvFSK：                                //FSK 数据接收器终止
            if(Event.dwParam == 4)                          //收到 1 帧 FSK 数据
            {
                unsigned char DataBuf[260];
                SsmGetRcvFSK(Event.nReference, DataBuf);//取出数据
                .....                                       //处理代码
            }
            else
            {
                .....                                       //接收失败的处理代码
            }
        break;
        .....
    }
}
```

2.16.2.3 SsmCheckRcvFSK

查询 FSK 数据接收任务的进展情况。

函数原型：

int SsmCheckRcvFSK(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	FSK 数据接收器尚未完成本次接收任务
1	因超时而终止
2	因收到结束符而终止
3	因收到指定长度数据而终止
4	因收到指定数据格式数据而终止
5	因应用程序调用SsmStopRcvFSK函数而终止

功能描述：

查询FSK数据接收任务的进展情况，该任务可以是由函数SsmStartRcvFSK或SsmStartRcvFSK_II启动。

注意事项：

- 如果应用程序使用基于事件的编程方式，建议使用 E_PROC_RcvFSK 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRcvFSK

2.16.2.4SsmStopRcvFSK

取消 FSK 数据接收任务。

函数原型：

int SsmStopRcvFSK(int ch)

参数说明：

ch	通道号

返回值：

-1	调用失败
0	调用成功

功能描述：

取消FSK数据接收任务。本函数可以取消由函数SsmStartRcvFSK或SsmStartRcvFSK_II启动的FSK数据接收任务。

注意事项：

- 驱动程序执行完本函数后，会向应用程序抛出 E_PROC_RcvFSK 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRcvFSK，SsmCheckRcvFSK

2.16.2.5SsmClearRcvFSKBuf

清空驱动程序内部的 FSK 数据接收缓冲区。

函数原型：

int SsmClearRcvFSKBuf(int ch)

参数说明：

ch	通道编号

返回值：

-1	出现错误，失败原因可以通过调用函数获得
0	成功

功能描述：

清空驱动程序内部的 FSK 数据接收缓冲区。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRcvFSK

2.16.2.6SsmGetRcvFSK

取出保存在驱动程序内部缓冲区中的 FSK 数据。

函数原型：

int SsmGetRcvFSK(int ch, PUCHAR pucBuf)

参数说明：

ch	通道编号
pucBuf	缓冲区指针，由应用程序分配存储空间，不能少于 260 字节

返回值：

-1	调用失败
≥0	收到的 FSK 数据的字节长度（不包括结束符）

功能描述：

取出保存在驱动程序内部缓冲区中的 FSK 数据。

注意事项：

- 每当应用程序调用 SsmStartRcvFSK 或启动接收任务时，驱动程序都会将保存 FSK 数据的内部缓冲区自动清空。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmCheckRcvFSK，SsmClearRcvFSKBuf

2.17 有关电压检测器的函数

2.17.1 忽略电压检测结果

2.17.1.1 SsmSetIgnoreLineVoltage

设置是否忽略线路上的电压检测结果。

函数原型：

int SsmSetIgnoreLineVoltage(int ch, BOOL bIgnore)

参数说明：

ch	通道编号
bIgnore	=FALSE：不忽略 =TRUE： 忽略

返回值：

-1	操作失败
0	操作成功

功能描述：

设置是否忽略线路上的电压检测结果。如果忽略线路上的电压检测结果，录音通道会一直保持“摘机”状态。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。

注意事项：

- 本函数只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）；
- 本函数设置的参数也可以通过配置项IgnoreLineVoltage进行设置，缺省值为不忽略。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIgnoreLineVoltage

2.17.1.2 SsmGetIgnoreLineVoltage

查询通道是否忽略线路上的电压检测结果。

函数原型：

int SsmGetIgnoreLineVoltage(int ch)

参数说明：

ch	通道编号

返回值：

-1	操作失败
0	不忽略
1	忽略

功能描述：

查询通道是否忽略线路上的电压检测结果。

注意事项：

- 本函数只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetIgnoreLineVoltage

2.17.2 设置检测摘机/挂机动作的门限电压

2.17.2.1 SsmSetDtrmLineVoltage

设置模拟中继线录音通道摘机/挂机的判定电压。

函数原型：

int SsmSetDtrmLineVoltage(int ch, WORD wDtrmValtage)

参数说明：

ch	通道编号

wDtrmValtage	判定电压，单位为伏特，取值范围：5~48

返回值：

-1	调用失败
0	调用成功

功能描述：

设置模拟中继线录音通道摘机/挂机的判定电压。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。

注意事项：

- 本函数只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）；
- 本函数设置的参数也可以通过配置项IsHangupDtrmVoltage进行设置，缺省值为 26V。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetDtrmLineVoltage

2.17.2.2SsmGetDtrmLineVoltage

获取模拟中继线录音通道摘机/挂机的判定电压。

函数原型：

int SsmGetDtrmLineVoltage(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	返回通道上的电话摘/挂机的判别电压，单位为伏特(V)

功能描述：

获取模拟中继线录音通道摘机/挂机的判定电压。

注意事项：

- 本函数只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetDtrmLineVoltage

2.17.3 获取电压检测器的输出结果

2.17.3.1 SsmGetLineVoltage

获取线路上的电压值。

函数原型：

int SsmGetLineVoltage(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	返回线路上的电压值，单位为伏特（V）

功能描述：

获取线路上的电压值。使用事件E_CHG_LineVoltage也可以用于获取线路的电压值。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。

注意事项：

- 本函数只适用于 ATP 系列板卡的录音通道和 SHT 系列板卡的模拟中继线通道；
- 模拟中继线上的电压值可能会在 1V-2V 的范围内波动，这是正常现象。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.17.4 有关极性反转检测器的函数

2.17.4.1 SsmGetPolarRvrsCount

获取通道自摘机以后累计检测到的极性反转次数。

函数原型：

int SsmGetPolarRvrsCount(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	极性反转的次数

功能描述：

获取通道自摘机以后累计检测到的极性反转次数。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。

注意事项：

- 本函数只适用于具有极性反转侦测的通道；
- 使用 E_CHG_PolarRvrsCount 事件也可以实现同样的功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：E_CHG_PolarRvrsCount

2.17.5 有关铃流信号检测器的函数

2.17.5.1 SsmGetRingCount

获取检测到的铃流信号次数。

函数原型：

int SsmGetRingCount(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	振铃计数

功能描述：

获取检测到的铃流信号次数。详细信息请参见第 1 章中的“模拟电话线的电压变化”部分内容。

注意事项：

- 事件 E_CHG_RingCount 也可实现相同功能。如果希望应用程序具有良好的可移植性，推荐使用事件 E_CHG_RingCount；
- 本函数只支持 SHT 系列的模拟中继线通道和 ATP 系列的模拟电话线录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmClearRingCount，SsmGetChState

2.17.5.2 SsmClearRingCount

清除保存在驱动程序内部的铃流信号的计数值。

函数原型：

int SsmClearRingCount (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

清除保存在驱动程序内部的铃流信号的计数值。详细信息请参见第 1 章中的“模拟电话线的电压变化”部分内容。

注意事项：

- 本函数只支持 SHT 系列的模拟中继线通道和 ATP 系列的模拟电话线录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetRingCount

2.18 有关数字中继线信息的函数

2.18.1 SsmGetMaxPcm

获取应用系统中的数字中继线总数。

函数原型：

int SsmGetMaxPcm()

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	系统中的数字中继线的总数，由[PcmInfo]配置节中的配置项TotalPcm决定。

功能描述：

获取应用系统中的数字中继线总数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPcmInfo

2.18.2 SsmGetPcmInfo

获取数字中继线接口单元（LIU）的工作参数。

函数原型：

int SsmGetPcmInfo(int nPcmNo, int* pnSSxMode, int* pnBoardId, int* pnBoardPcmNo, int* pnUsePcmTS16, PDWORD pdwRcvrMode, PDWORD pdwEnableAutoCall, PDWORD pdwAutoCallDirection)

参数说明：

nPcmNo	数字中继线的逻辑编号，取值范围：0~N-1，N为[PcmInfo]配置节中配置项TotalPcm的设定值，可通过函数SsmGetMaxPcm获得。
pnBoardId	返回数字中继线所在板卡的 ID 号
pnBoardPcmNo	返回数字中继线在板卡上的内部编号
pnSSxMode	返回 PCM 上使用的信令方式： 0： ISDN 信令 1： 中国 1 号信令（简称 SS1） 7： No. 7 号信令（简称 SS7）
pnUsePcmTS16	返回 PCM 的 16 时隙是否作为话路使用。 1：是 0：不是
PdwRcvrMode	返回数字中继线接口单元（LIU）的工作模式，包含 32 比特，各比特定义如下： Bit1,Bit0： 时钟设置，取值范围： 00：线路同步方式、主时钟 01：自振荡方式、主时钟 10：从时钟 11：保留

	Bit2：线路的接入方式 0：120 欧姆双绞线 1：75 欧姆同轴电缆 Bit3：LIU 是否对 16 时隙进行消抖处理 0：不进行消抖处理（缺省值） 1：进行 14ms 的消抖处理 Bit4：设置语音信道的隔位反转功能（ADI） 0：关闭（缺省值） 1：开启 Bit6,Bit5：设置 LIU 的接收均衡器增益 00：由 LIU 自动控制(缺省值) 01： 0 DB 10：12 DB 11： 6 DB Bit7：保留 Bit15~Bit8：设置 LIU 的消除抖动的控制字 Bit31~Bit16：保留
pdwEnableAutoCall	返回 32 个时隙的自动呼叫控制字（32 比特），最高位比特（bit31）对应时隙 0，最低位比特（bit0）对应于时隙 31。比特为 1 表示允许自动呼叫，为 0 表示禁止自动呼叫。
pdwAutoCallDirection	返回 32 个时隙的自动呼叫方向，一个比特对应一个时隙，最高位比特（bit31）对应时隙 0，最低位比特（bit0）对应时隙 31。比特值为 0 表示只允许呼入，为 1 表示只允许呼出。

返回值：

-1	调用失败
0	调用成功

功能描述：

获取数字中继线接口单元（LIU）的工作参数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPcmLinkStatus

2.18.3 SsmGetPcmLinkStatus

获取数字中继线的同步状态。

函数原型：

int SsmGetPcmLinkStatus(int nPcmNo, PWORD pwPcmLinkStatus)

参数说明：

nPcmNo	数字中继线的逻辑编号，取值范围：0~N-1，N为[PcmInfo]配置节中配置项TotalPcm的设定值，可通过函数SsmGetMaxPcm获得。
pwPcmLinkStatus	返回数字中继线的同步状态，共 16 个比特，bit 0 为最低有效位。比特值为 0 表示工作正常，为 1 时： bit0=1： 基本帧同步丢失 bit1=1： 基本帧同步丢失时间大于 100ms bit2=1： 复帧同步丢失（SS1 only） bit3=1： CRC 复帧同步丢失（SS1 only） bit4=1： LIU 收到对端奇帧失步对告信号（SS1 only） bit5=1： LIU 收到对端全 1 告警信息（SS1 only）

	bit6=1： LIU 收到对端 16 时隙全 1 告警信息（SS1 only） bit7=1： PCM 信号丢失（SS1 only） bit8=1： PCM 链路收到对端辅助告警(AUXiliary alarm Pattern)(除 8E1/16E1 的 SS1 板卡)，或者同步丢失（8E1/16E1） bit9=1： PCM 链路收到对端复祯失步告警（除 8E1/16E1 的 SS1 板卡），或者载波丢失（8E1/16E1） bit10=1：收到对端告警（8E1/16E1） 其余比特：保留，全部为 0

返回值：

-1	调用失败
0	调用成功

功能描述：

获取数字中继线的同步状态。

注意事项：

- 对于中国 No.1 信令，有的交换机支持 CRC 复帧校验，有的不支持。因此，当出现 CRC 复帧同步丢失时不一定表明线路同步出现故障；
- 对于 SS7 信令和 ISDN 信令，16 时隙通常用于传输共路信令，没有复帧和 CRC 复帧的概念，因此，只要基本帧同步信息正常（bit0 和 bit1 均为 0），就可以正常使用。另外，由于 SS7 信令和 ISDN 信令均为为数字信令，具有一定的容错能力，允许基本帧同步出现小范围的抖动。所以，如果将本函数返回的线路同步状态用于告警等目的，通常以 bit0 从 0 变化为 1 作为告警依据。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPcmInfo

2.18.4 SsmPcmTsToCh

根据 PCM 的逻辑编号和时隙编号查询对应的通道编号。

函数原型：

int SsmPcmTsToCh(int nLocalPcmNo, int nTs)

参数说明：

nLocalPcmNo	数字中继线的逻辑编号。有关数字中继线的详细内容请参见第 1 章中”基本概念”部分
nTs	时隙编号，取值范围：1~31

返回值：

-1	操作失败
≥0	对应通道的逻辑编号

功能描述：

根据 PCM 的逻辑编号和时隙编号查询对应的通道编号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：SsmChToPcmTs

2.18.5 SsmChToPcmTs

由通道编号查询对应的 PCM 逻辑编号和时隙编号。

函数原型：

int SsmChToPcmTs(int ch, int* pnLocalPcmNo, int* pnTs)

参数说明：

ch	通道编号
pnLocalPcmNo	返回通道 ch 所在的 PCM 逻辑编号。
pnTs	通道 ch 所在的时隙编号

返回值：

-1	操作失败
≥0	返回 pnLocalPcmNo 参数中的值

功能描述：

由通道编号查询对应的 PCM 逻辑编号和时隙编号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPcmTsToCh

2.18.6 SsmSetPcmClockMode

设置数字中继线的时钟模式。

函数原型：

int SsmSetPcmClockMode(int nPcmNo, int nClockMode)

参数说明：

nPcmNo	数字中继线的逻辑编号，取值范围：0~N-1（N为配置项TotalPcm的设定值，可以通过函数SsmGetMaxPcm获取）。
nClockMode	时钟模式，取值范围： 0：主时钟、线路同步方式（板卡必须为主卡） 1：主时钟、自振荡方式（板卡必须为主卡） 2：从时钟

返回值：

-1	调用失败
0	调用成功

功能描述：

设置数字中继线的时钟模式。更多信息请参见第 1 章中“设置应用系统的时钟”部分内容。

注意事项：

- 本函数只适用于 SHD、DTP 系列板卡；
- 配置项PcmClockMode可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPcmInfo

2.18.7 SsmGetInboundLinkSet

来话呼叫时，获取来自对端交换机的 TUP/ISUP 消息所使用的链路组信息。

函数原型：

int SsmGetInboundLinkSet(int ch, LPWORD pwLinkSetNo, LPSTR pszOpc, LPSTR pszDpc)

参数说明：

ch	通道号
pwLinkSetNo	链路组编号
pszOpc	OPC
pszDpc	DPC

返回值：

1	调用成功
-1	调用失败

功能描述：

来话呼叫时，获取来自对端交换机的 TUP/ISUP 消息所使用的链路组信息，包括链路组的编号、DPC、OPC。

注意事项：

- 本函数只适用于 TUP/ISUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.19 SHT 系列板卡专用函数（CTI 系列）

2.19.1 设置联合模块工作模式

2.19.1.1 SsmSetUnimoduleState

设置是否将联合模块上的模拟中继线通道直接连接到坐席通道。

函数原型：

int SsmSetUnimoduleState(int ch,int nState)

参数说明：

ch	联合模块中模拟中继线通道的编号
nState	=0：模拟中继线通道直接连接到坐席通道 =1：模拟中继线通道与坐席通道没有任何联系，相当于两个完全独立控制的通道。

返回值：

-1	失败
0	成功

功能描述：

设置是否将联合模块上的模拟中继线通道直接连接到坐席通道。

注意事项：

- 配置项UnimoduleState可以实现相同功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.19.2 模拟中继通道专用函数

2.19.2.1 在模拟线路上产生闪断信号

2.19.2.1.1SsmTxFlash

在模拟中继线路上产生一次闪断（相当于话机的拍叉簧动作）。

函数原型：

int SsmTxFlash(int ch, WORD time)

参数说明：

ch	通道编号
time	设置闪断的保持时间，单位为毫秒，取值范围：32~1000，通常设置为 500ms

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

在模拟中继线路上产生一次闪断，即模拟话机的拍叉簧动作。

当驱动程序完成一次产生闪断的操作后，会向应用程序抛出E_PROC_SendFlash事件。操作执行的进展情况也可以用函数SsmChkTxFlash取得。

注意事项：

- 本函数只支持 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmChkTxFlash

2.19.2.1.2SsmSetTxFlashCharTime

设置用函数 SsmTxDtmf 在模拟中继线上产生的闪断的保持时间。

函数原型：

int SsmSetTxFlashCharTime（int ch,WORD time）

参数说明：

ch	通道编号
Time	闪断信号的保持时间，单位为毫秒，取值范围：32~1000，默认值为 500

返回值：

0	操作成功
-1	操作失败。失败原因可以通过调用函数SsmGetLastErrMsg获得

功能描述：

使用'!'字符作为参数调用函数 SsmTxDtmf 也可以在模拟中继线上产生一次闪断，本函数用于设置该字符产生的闪断信号的保持时间。

注意事项：

- 配置项DefaultTxFlashTime可以实现相同功能，缺省值为 500ms。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetTxFlashCharTime，SsmTxDtmf

2.19.2.1.3SsmGetTxFlashCharTime

获取用函数 SsmTxDtmf 在模拟中继线上产生的闪断的保持时间。

函数原型：

int SsmGetTxFlashCharTime(int ch,WORD time)

参数说明：

ch	通道编号

返回值：

>0	闪断信号的保持时间，单位为毫秒
-1	操作失败。失败原因可以通过调用函数SsmGetLastErrMsg获得

功能描述：

获得调用函数 SsmTxDtmf 产生的闪断信号的保持时间。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetTxFlashCharTime

2.19.2.1.4SsmChkTxFlash

查询在模拟中继线上产生闪断信号的操作是否完成。

函数原型：

int SsmChkTxFlash(int ch)

参数说明：

ch	通道编号

返回值：

-1	出现错误
0	产生闪断的操作已经完成
1	产生闪断的操作正在进行

功能描述：

查询在模拟中继线上产生闪断信号的操作是否完成。

注意事项：

- 只有模拟中继线通道才支持产生闪断信号的操作。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmTxFlash

2.19.2.1.5SsmTxFlashEx

在模拟中继线路上产生一次闪断（相当于话机的拍叉簧动作），同时设置闪断后通道的当前状态。

函数原型：

int WINAPI SsmTxFlashEx(int ch, WORD time, int nChState, BOOL bIgnoreDlTn)

参数说明：

ch	通道编号
time	设置闪断的保持时间，单位为毫秒，取值范围：32~1000，通常设置为 500ms
nChState	设置闪断后通道的状态，取值范围：S_CALL_PICKUPED(1)、S_CALL_TALKING(3)
bIgnoreDlTn	设置闪断后是否忽略拨号音，取值范围：FALSE（不忽略）、TRUE（忽略）

返回值：

-1	调用失败。失败原因可以通过调用函数SsmGetLastErrMsg获得
0	调用成功

功能描述：

在指定通道上产生一次闪断（拍叉簧）动作，闪断持续的时间由参数 time 指定。同时设置闪断后通道的当前状态。

注意事项：

- 本函数只支持 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmTxFlash、SsmChkTxFlash

2.19.2.2 通道状态的强制迁移

2.19.2.2.1 SsmSetChState

强制模拟中继线通道的状态迁移到通话状态。

函数原型：

int SsmSetChState (int ch, int nState)

参数说明：

ch	通道编号
nState	通道的新状态，必须设置为 3。

返回值：

-1	调用失败
0	调用成功

功能描述：

当模拟中继线通道由于某种意外发生挂起时，本函数可以强制模拟中继线通道的状态迁移到通话状态。

注意事项：

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。只有在某些特殊的应用场合，例如应用程序发现通道非正常挂起后，希望能将通道恢复为通话状态时，才会用到本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.19.2.3 去话接续时检测被叫摘机的函数

2.19.2.3.1 SsmStartPickupAnalyze

启动“增强的远端摘机检测器”。

函数原型：

int SsmStartPickupAnalyze(int ch)

参数说明：

ch	通道号

返回值：

-1	失败
0	成功

功能描述：

启动“增强的远端摘机检测器”。 更多信息请参见第 1 章中“增强的远端摘机检测器”部分内容。

注意事项：

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.5.2.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPickup

2.19.2.3.2SsmSetCalleeHookDetectP

设置模拟中继线通道上的“增强的远端摘机检测器”的工作参数。

函数原型：

int SsmSetCalleeHookDetectP(int ch, WORD wMulti, WORD wValidTime)

参数说明：

ch	通道号
wMulti	灵敏度，更多信息请参见配置项HookEngyConfigMulti的说明。
wValidTime	最小保持时间，更多信息请参见配置项HookValidEngyCnt的说明。

返回值：

-1	失败
0	成功

功能描述：

设置模拟中继线通道上的“增强的远端摘机检测器”的工作参数。更多信息请参见第 1 章中“增强的远端摘机检测器”部分内容。

注意事项：

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.5.2.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartPickupAnalyze

2.19.2.3.3SsmGetPickup

查询模拟中继线通道上“增强的远端摘机检测器”的检测结果。

函数原型：

int SsmGetPickup (int ch)

参数说明：

ch	通道号

返回值：

-1	失败
0	被叫用户没有摘机
1	被叫用户已经摘机

功能描述：

查询模拟中继线通道上“增强的远端摘机检测器”的检测结果。更多信息请参见第 1 章中“增强的远端摘机检测器”部分内容。

注意事项：

- 使用 E_SYS_RemotePickup 事件可以实现相同功能（推荐使用）；
- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.5.2.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartPickupAnalyze，SsmSetCalleeHookDetectP

2.19.2.4检测通道摘机执行情况的函数

2.19.2.4.1SsmCheckActualPickup

查询摘机操作是否执行完毕。

函数原型：

int SsmCheckActualPickup(int ch)

参数说明：

ch	通道编号

返回值：

-1	操作失败
0	指定通道处于实际挂机状态
1	指定通道处于实际摘机状态

功能描述：

查询摘机操作是否执行完毕。更多信息请参见函数SsmPickup的说明。

注意事项：

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmPickup

2.19.3 坐席通道专用函数

2.19.3.1 向电话机发送振铃信号

2.19.3.1.1 SsmStartRing

2.19.3.1.2 SsmStartRingWithCIDStr

2.19.3.1.3 SsmStartRingWithFskCID

向电话机发送振铃信号。SsmStartRing 只是产生振铃信号，SsmStartRingWithCIDStr 和 SsmStartRingWithFskCID 还可以向话机发送主叫号码信息。

函数原型：

int SsmStartRing(int ch)

int SsmStartRingWithCIDStr(int ch, LPSTR pCIDBuf, DWORD dwCIDLen, DWORD dwDelayTime)

int SsmStartRingWithFskCID(int ch, LPSTR pFSKStream, DWORD dwMaxBit, DWORD dwDelayTime)

参数说明：

ch	通道编号
pCIDBuf	存放主叫号码信息的缓冲区首地址。例如：”057188861158”
dwCIDLen	pCIDBuf 中主叫号码字符的个数，取值范围：1~20
pFSKStream	存放调制后的FSK比特流的缓冲区首地址。每个字节的低位比特（Bit 0）最先发送，高位比特（Bit 7）最后发送。函数SsmTransFskData可以用来将ASCII字符形式的主叫号码信息调制成FSK比特流。
dwMaxBit	pFSKStream 中的比特总数。
dwDelayTime	从第 1 个振铃信号的下降沿开始，到线路上出现第 1 个 FSK 数据的时间间隔，单位为毫秒，取值范围：500~1500。 注：FSK 方式的主叫号码在第 1 个与第 2 个振铃信号之间传送。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

向电话机发送振铃信号。

函数SsmStopRing可以用来随时终止向电话机发送振铃信号。

注意事项：

- 本函数只适用于 SHT 系列板卡的坐席通道，SsmStartRing 函数可以适用于坐席通道和磁石话机通道，SsmStartRingWithCIDStr、SsmStartRingWithFskCID 只适用于坐席通道；
- 如果使用 SsmStartRingWithCIDStr 或 SsmStartRingWithFskCID 函数，在函数执行期间，不能调用任何语音播放的函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopRing

2.19.3.1.4fPcm_ConvertFskCID

将主叫号码信息用 FSK 方式调制成适合于在模拟电话线上传输的音频信号。

函数原型：

DWORD fPcm_ConvertFskCID(LPSTR pFskStream, int nFskLen, LPSTR pszCIDNumber, LPSTR pszTime, LPSTR pszName, int nMode)

参数说明：

pFskStream	保存调制后的 FSK 比特流的缓冲区首地址，输出参数，由应用程序分配存储空间，空间应不少于 500 字节。
nFskLen	pFskStream 缓冲区的大小，单位为字节
pszCIDNumber	存放主叫号码的缓冲区首地址，字符个数应在 4~14 之间
pszTime	存放时间信息的缓冲区首地址，长度为 8 字节，格式为“MMDDhhmm”，MM 为月，DD 为日，hh 为时，mm 为分。例如“12041224”表示 12 月 4 日 12 点 24 分。 如果 pszTime 为 NULL，驱动程序会使用系统时间来填充。
pszName	存放附加信息的缓冲区首地址，长度为 4~14 字节，可以为空。 注：大多数电话机不能显示附加信息
nMode	FSK 主叫号码的帧格式，0 表示复帧，1 表示单帧

返回值：

0	调制失败
>0	返回调制后总比特数

功能描述：

将主叫号码信息用 FSK 方式调制成适合于在模拟电话线上传输的音频信号。

注意事项：

- 本函数通常与函数SsmStartRingWithFskCID配合使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartRingWithFskCID

2.19.3.1.5SsmStopRing

停止向坐席通道或磁石话机通道发送振铃信号。

函数原型：

int SsmStopRing(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

停止向坐席通道或磁石话机通道发送振铃信号。发送振铃信号的操作可以是由函数SsmStartRing、SsmStartRingWithCIDStr或SsmStartRingWithFskCID发起。

注意事项：

- 本函数只适用于 SHT 系列板卡的坐席通道和磁石话机通道；
- 本函数调用后，保存在驱动程序内部的振铃信号计数器将被清零。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartRing， SsmStartRingWithCIDStr，SsmStartRingWithFskCID

2.19.3.1.6SsmCheckSendRing

获取坐席通道和磁石话机通道的铃流发生器的进展情况。

函数原型：

int SsmCheckSendRing(int ch,int * pnCnt)

参数说明：

ch	通道编号
pnCnt	返回保存在驱动程序内部的振铃信号计数器值

返回值：

-1	调用失败
0	通道上没有发送振铃信号
1	通道上正在发送振铃信号

功能描述：

获取坐席通道和磁石话机通道的铃流发生器的进展情况。

注意事项：

- 本函数只适用于 SHT 系列板卡的坐席通道和磁石话机通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStartRing，SsmStartRingWithCIDStr，SsmStartRingWithFskCID

2.19.3.2检测到话机摘机后自动送拨号音

2.19.3.2.1SsmSetASDT

设置当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的功能。

函数原型：

int SsmSetASDT(int ch, BOOL bEnAutoSendDialTone)

参数说明：

ch	通道编号
bEnAutoSendDialTone	=TRUE： 开启 =FALSE：关闭

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的功能。

注意事项：

- 本函数只适用于 SHT 系列板卡的坐席通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmGetASDT

2.19.3.2.2SsmGetASDT

获取当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的标志。

函数原型：

int SsmGetASDT(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	状态关
1	状态开

功能描述：

获取当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的标志。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetASDT

2.19.3.3 检测话机的闪断信号

2.19.3.3.1 SsmSetLocalFlashTime

设置判断闪断信号的最大保持时间。

函数原型：

int SsmSetLocalFlashTime(int nFlashTime)

参数说明：

nFlashTime	闪断信号的最大保持时间，单位为毫秒，取值范围：32~2000

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得。
0	调用成功

功能描述：

设置判断闪断信号的最大保持时间。详细信息请参见第 1 章中“检测话机的闪断信号”部分内容。

注意事项：

- 本函数只适用于坐席通道；
- 配置项MaxLocalFlashTime也可实现相同通能，缺省值为 700 毫秒。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.19.3.3.2 SsmGetFlashCount

获取坐席通道上保存在驱动程序内部的闪断信号计数器的值。

函数原型：

int SsmGetFlashCount(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	闪断信号计数器的值

功能描述：

获取坐席通道上保存在驱动程序内部的闪断信号计数器的值。

注意事项：

- 本函数只适用于坐席通道；
- 坐席通道在进入摘机状态或挂机状态时驱动程序自动将闪断计数器清零。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmClearFlashCount

2.19.3.3.3 SsmClearFlashCount

清零指定坐席通道上的闪断计数器。

函数原型：

int SsmClearFlashCount (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

清零指定坐席通道上的闪断计数器。

注意事项：

- 本函数只适用于坐席通道；
- 坐席通道在进入摘机状态或挂机状态时驱动程序自动将闪断计数器清零。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetFlashCount

2.19.3.4 检测话机的摘机/挂机动作

2.19.3.4.1 SsmGetHookState

获取坐席通道的摘机/挂机状态。

函数原型：

int SsmGetHookState(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	挂机状态
1	摘机状态

功能描述：

获取坐席通道的摘机/挂机状态。使用 E_CHG_HookState 事件也可以实现同样功能。

注意事项：

- 本函数只适用于 SHT 系列板卡的坐席通道；
- 当坐席通道迁移到“摘机”状态或“挂机”状态时，驱动程序会将内部的闪断计数器自动清零。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：E_CHG_HookState

2.19.3.5 在电话线上产生极性反转信号

2.19.3.5.1 SsmSetPolarState

设置与坐席通道相连的模拟电话线的馈电电压极性。

函数原型：

int SsmSetPolarState(int ch, int nPolar)

参数说明：

ch	通道编号
nPolar	要设置的通道极性。取值范围：0 或 1

返回值：

-1	调用失败
0	调用成功

功能描述：

设置与坐席通道相连的模拟电话线的馈电电压极性。通道的当前电压极性可以通过函数SsmGetPolarState获得。详细信息请参见第 1 章中“在电话线上产生极性反转信号”部分内容。

注意事项：

- 只有在配置项UserSendPolar的设定值为 1 时，才能调用本函数；
- 本函数只适用于 SHT 系列板卡的坐席通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetPolarState

代码范例：

```
int v = SsmGetPolarState(ch);
SsmSetPolarState(ch, ~v);
```

2.19.3.5.2 SsmGetPolarState

获取与坐席通道相连的模拟电话线的馈电电压极性。

函数原型：

int SsmGetPolarState(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0,1	返回当前的极性状态

功能描述：

获取与坐席通道相连的模拟电话线的馈电电压极性。

注意事项：

- 本函数只适用于 SHT 系列板卡的坐席通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetPolarState

2.19.4 有关远端摘机检测器的函数（SHT 系列 only）

2.19.4.1 有关普通远端摘机检测器的函数

2.19.4.1.1 设置工作参数

2.19.4.1.1.1 SsmSetVoiceOnDetermineTime

设置语音活动信号的最小保持时间。

函数原型：

int SsmSetVoiceOnDetermineTime(int ch, WORD wIsVocDtrTime)

参数说明：

ch	通道编号
wIsVocDtrTime	最小保持时间，单位为毫秒，取值范围：16ms~1024ms。

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

设置语音活动信号的最小保持时间。只有当线路上连续出现语音活动信号，并且保持的时间大于参数wIsVocDtrTime的设定值时，驱动程序才会确认线路上出现了真正的语音活动信号。更多信息请参见第 1 章中“普通的远端摘机检测器”部分内容。

注意事项：

- 配置项VoiceOnDetermineTime可以实现相同功能；
- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetVoiceOnDetermineTime

2.19.4.1.2获取工作参数

2.19.4.1.2.1 SsmGetVoiceOnDetermineTime

获取语音活动信号的最小保持时间。

函数原型：

int SsmGetVoiceOnDetermineTime(int ch, PWORD pwIsVocDtrTime)

参数说明：

ch	通道编号
pwIsVocDtrTime	返回最小保持时间，单位为毫秒

返回值：

-1	调用失败，失败原因可以从函数SsmGetLastErrMsg取得
0	调用成功

功能描述：

取得驱动程序判定线路上出现话音信号的最小判决时间。

注意事项：

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetVoiceOnDetermineTime

2.20 SHD 系列板卡的高级编程 API（CTI 系列）

2.20.1 通道的闭塞控制（CTI 系列）

2.20.1.1闭塞本端的函数

2.20.1.1.1闭塞单个通道

2.20.1.1.1.1 SsmBlockLocalCh

闭塞本端通道，以禁止其进行去话呼叫，但仍然可以处理来话呼叫(TUP 通道)；或来话及去话呼叫均禁止（ISDN 通道）。

函数原型：

int SsmBlockLocalCh(int ch)

参数说明：

ch	输入。通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

闭塞本端通道。通道被闭塞后，对于TUP通道，不能进行去话呼叫，但仍然可以处理来话呼叫；对于ISDN通道，来话及去话呼叫均不支持。有关通道闭塞的详细内容请参见第 1 章中“通道的闭塞与解除”部分内容。

本函数适用于TUP通道及ISDN通道，会对通道的状态机产生影响。有关TUP及ISDN通道的状态机的详细说明请参见第 1 章中“TUP通道的状态转移”和“ISDN通道的状态转移图”部分内容（本函数触发的事件在状态机迁移图中的名称为Blocking）。

注意事项：

- 本函数只适用于 TUP 通道及 ISDN 通道；
- 本函数可以在任意时刻调用，但当通道上正在执行 AutoDial 任务时，本函数会导致任务立即失败。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnblockLocalCh，SsmBlockLocalPCM，SsmQueryLocalChBlockState，SsmBlockRemoteCh，SsmBlockRemotePCM

2.20.1.1.1.2 SsmUnblockLocalCh

解除对本端通道闭塞。

函数原型：

int SsmUnblockLocalCh(int ch)

参数说明：

ch	输入。通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

本函数解除对本端通道闭塞，包括因应用程序调用函数SsmBlockLocalCh或SsmBlockLocalPCM导致的某个通道的本地闭塞。

注意事项：

- 本函数只适用于 TUP 通道及 ISDN 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockLocalCh，SsmUnblockLocalPCM，SsmQueryLocalChBlockState

2.20.1.1.1.3 SsmQueryLocalChBlockState

获取通道的本端闭塞状态。

函数原型：

int SsmQueryLocalChBlockState(int ch, PDWORD pdwBlockState)

参数说明：

ch	通道编号
pdwBlockState	返回通道的本端闭塞状态。比特为 1 表示闭塞，0 表示未闭塞。各比特的含义为： Bit 0：因应用程序调用SsmBlockLocalCh函数而闭塞 Bit 1：因收到对端电路闭塞消息 BLO 而闭塞 Bit 2：因收到对端电路群闭塞消息 SGB 而闭塞 Bit 3：因收到对端电路群闭塞消息 HGB 而闭塞 Bit 4：因收到对端电路群闭塞消息 MGB 而闭塞 Bit 5：因应用程序调用SsmBlockLocalPCM函数而闭塞 Bit31~Bit6：保留

返回值：

-1	调用失败
0	调用成功

功能描述：

本函数获取通道是否被本端闭塞，以及闭塞的具体原因。

注意事项：

- 本函数只适用于 TUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockLocalCh，SsmBlockLocalPCM

2.20.1.1.2 闭塞整条数字中继线

2.20.1.1.2.1 SsmBlockLocalPCM

闭塞本端电路群（整条数字中继线）。

函数原型：

int SsmBlockLocalPCM(int nLocalPcmNo)

参数说明：

nLocalPcmNo	PCM 的逻辑编号

返回值：

-1	调用失败
0	调用成功

功能描述：

本函数闭塞本端整条数字中继线，使其包含的本端全部通道都不能进行去话呼叫，但仍然能够处理来话呼叫，本函数的执行效果相当于对该PCM中的全部通道调用SsmBlockLocalCh。有关本函数的更多信息请参见函数

SsmBlockLocalCh的说明。

注意事项：

- 本函数只适用于 TUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnblockLocalPCM，SsmQueryLocalPCMBlockState，SsmBlockLocalCh

2.20.1.1.2.2 SsmUnblockLocalPCM

解除对本端整条数字中继线的闭塞。

函数原型：

int SsmUnblockLocalPCM(int nLocalPcmNo)

参数说明：

nLocalPcmNo	PCM 的逻辑编号

返回值：

-1	调用失败
0	调用成功

功能描述：

本函数解除数字中继线上由于应用程序调用函数SsmBlockLocalPCM而引起的闭塞。本函数的执行效果相当于对该数字中继线所包含的所有通道调用SsmUnblockLocalCh。因此，有关本函数的情况可参见函数SsmUnblockLocalCh的说明。

注意事项：

- 本函数只适用于 TUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockLocalPCM，SsmQueryLocalPCMBlockState，SsmUnblockLocalCh

2.20.1.1.2.3 SsmQueryLocalPCMBlockState

获取数字中继线的本地闭塞状态。

函数原型：

int SsmQueryLocalPCMBlockState(int nLocalPcmNo, PDWORD pdwBlockState)

参数说明：

nLocalPcmNo	PCM 的逻辑编号
pdwBlockState	返回通道的本端闭塞状态。比特为 1 表示闭塞，0 表示未闭塞。各比特的含义为： Bit1~Bit0：保留 Bit 2：因收到对端交换机的电路群闭塞消息 SGB 而闭塞 Bit 3：因收到对端交换机的电路群闭塞消息 HGB 而闭塞 Bit 4：因收到对端交换机的电路群闭塞消息 MGB 而闭塞

	Bit 5	因应用程序调用SsmBlockLocalPCM函数而闭塞
	Bit31~Bit6	保留

返回值：

-1	调用失败
0	调用成功

功能描述：

本函数获取数字中继线的本地闭塞状态。

注意事项：

- 本函数只适用于 TUP 通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockLocalCh，SsmBlockLocalPCM

2.20.1.2 闭塞对端的函数

2.20.1.2.1 闭塞单个通道

2.20.1.2.1.1 SsmQueryOpBlockRemoteCh

查询通道是否支持闭塞对端的操作。

函数原型：

int SsmQueryOpBlockRemoteCh(int ch)

参数说明：

ch	输入。通道编号

返回值：

-1	调用失败
0	指定通道不支持闭塞对端的操作
1	指定通道支持闭塞对端的操作

功能描述：

查询通道是否支持闭塞对端通道的操作。

注意事项：

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端通道的操作。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.20.1.2.1.2 SsmBlockRemoteCh

闭塞对端通道。

函数原型：

int SsmBlockRemoteCh(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

闭塞对端通道，以阻止对端交换机在此通道上向本端发起呼叫，但本端仍可以向对端交换机发起去话呼叫。对端通道是指对端交换机中具有与本端通道相同的 CIC 编号的电路。当对端通道进入闭塞状态后，其去话呼叫被禁止，本端通道上就可以保证没有来话呼叫。闭塞对端通道的功能通常用于系统维护等目的。

闭塞对端通道的消息处理过程为：

(1) 驱动程序向对端交换机发送电路闭塞消息 BLO；

(2) 对端交换机收到 BLO 消息后，应答闭塞证实信号 BLA；

(3) 收到对端交换机的 BLA 消息后，驱动程序向应用程序抛出 E_CHG_RemoteChBlock 事件，闭塞对端通道操作完成。

函数SsmGetRemoteChBlockStatus也可以用来检查闭塞对端通道的操作是否完成。

注意事项：

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端的操作。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnblockRemoteCh，SsmGetRemoteChBlockStatus

2.20.1.2.1.3 SsmUnblockRemoteCh

解除对对端通道的闭塞。

函数原型：

int SsmUnblockRemoteCh(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

解除对对端通道的闭塞，使对端交换机恢复在此通道上向本端发起呼叫的能力。解除闭塞对端通道的消息处理过程为：

(1) 驱动程序向对端交换机发送电路解除闭塞消息 UBL；

(2) 对端交换机收到 UBL 消息后，应答解除闭塞证实信号 UBA；

(3) 收到对端交换机的 UBA 消息后，驱动程序向应用程序抛出 E_CHG_RemoteChBlock 事件，解除闭塞对端通道操作完成。

函数SsmGetRemoteChBlockStatus也可以用来检查解除闭塞对端通道的操作是否完成。

注意事项：

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端的操作。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockRemoteCh，SsmGetRemoteChBlockStatus，SsmBlockRemotePCM

2.20.1.2.1.4 SsmGetRemoteChBlockStatus

查询对端交换机上的通道是否被本端闭塞。

函数原型：

int SsmGetRemoteChBlockStatus(int ch)

参数说明：

ch	通道编号，输入

返回值：

-1	调用失败
0	没有被本端闭塞，或已经成功解除由本端导致的闭塞
1	已经成功闭塞对端
2	已经发出电路闭塞消息，正在等待对端交换机的闭塞证实信号
3	已经发出电路解除闭塞消息，正在等待对端交换机的解除闭塞证实信号

功能描述：

查询对端交换机上的通道是否被本端闭塞，返回结果可以用来检查函数SsmBlockRemoteCh或SsmUnblockRemoteCh的执行进展情况。

注意事项：

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端的操作。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockRemoteCh，SsmUnblockRemoteCh

2.20.1.2.2 闭塞整条数字中继线

2.20.1.2.2.1 SsmBlockRemotePCM

闭塞对端电路群。

函数原型：

int SsmBlockRemotePCM(int nLocalPcmNo, DWORD dwBlockMode)

参数说明：

nLocalPcmNo	PCM 逻辑编号
dwBlockMode	选择向对端交换机发送的电路群闭塞消息。取值范围为： TUP 协议： 1：发送面向维护的群闭塞消息 MGB； 2：发送面向硬件故障的群闭塞消息 HGB； 3：发送软件产生的群闭塞消息 SGB； ISUP 协议： 0：发送面向维护的群闭塞消息 CGB； 1：发送面向硬件故障的群闭塞消息 CGB；

返回值：

-1	调用失败
0	调用成功

功能描述：

用电路群闭塞消息将对端交换机的整条数字中继线闭塞，使对端交换机中包含在此 PCM 上的全部通道都不能进行去话呼叫，只能处理来话呼叫。本函数的执行效果相当于对数字中继线上的全部通道同时调用 SsmBlockRemoteCh。

闭塞对端电路群的消息处理过程为：

(1) 驱动程序根据 dwBlockMode 参数的设置，向对端交换机发送指定的电路群闭塞消息；

(2) 对端交换机收到电路群闭塞消息后，应答电路群闭塞证实消息（TUP 协议的电路群闭塞消息 MGB、HGB、SGB 所对应的电路群闭塞证实消息分别为 MBA、HBA、SBA，ISUP 协议的电路群闭塞消息 CGB 所对应的电路群闭塞证实消息为 CGBA）；

(3) 收到对端交换机的电路群闭塞证实消息后，驱动程序向应用程序抛出 E_CHG_RemotePCMBlock 事件，闭塞对端电路群的操作完成。

函数SsmGetRemotePCMBlockStatus也可以用来检查对对端电路群的解除闭塞操作是否完成。

注意事项：

- 本函数只支持 TUP 协议和 ISUP 协议；
- 按照 TUP 协议和 ISUP 协议的规定，电路群闭塞消息需要发送 2 次才有效，所以驱动程序会向对端交换机连续发送 2 次电路群闭塞消息，因此应用程序只需调用本函数一次。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnblockRemotePCM，SsmGetRemotePCMBlockStatus

相关事件：E_CHG_RemotePCMBlock

2.20.1.2.2.2 SsmUnblockRemotePCM

解除闭塞对端电路群。

函数原型：

int SsmUnblockRemotePCM(int nLocalPcmNo, DWORD dwUnblockMode)

参数说明：

nLocalPcmNo	PCM 逻辑编号

dwBlockMode	选择向对端交换机发送的电路群解除闭塞消息。取值范围为： TUP 协议： 1：发送 MGU，以解除因 MGB 消息而引起的电路群闭塞； 2：发送 HGU，以解除因 HGB 消息而引起的电路群闭塞； 3：发送 SGU，以解除因 SGB 消息而引起的电路群闭塞。 ISUP 协议： 0：发送面向维护的群解除闭塞消息 CGU； 1：发送面向硬件故障的群解除闭塞消息 CGU。

返回值：

-1	调用失败
0	调用成功

功能描述：

解除对对端交换机的整条数字中继线闭塞，使对端交换机中包含在此数字中继线上的全部通道恢复去话呼叫的能力。本函数的执行效果相当于对数字中继线上的全部通道同时调用 SsmUnblockRemoteCh。

闭塞对端电路群的消息处理过程为：

（1） 驱动程序根据 dwBlockMode 参数的设置，向对端交换机发送指定的电路群解除闭塞消息；

（2） 对端交换机收到电路群解除闭塞消息后，应答电路群解除闭塞证实消息（TUP 协议的电路群解除闭塞消息 MGU、HGU、SGU 所对应的电路群解除闭塞证实消息分别为 MUA、HUA、SUA，ISUP 协议的电路群解除闭塞消息 CGU 所对应的电路群解除闭塞证实消息为 CGUA）；

（3） 收到对端交换机的电路群解除闭塞证实消息后，驱动程序向应用程序抛出 E_CHG_RemotePCMBlock 事件，解除闭塞对端电路群的操作完成。

函数SsmGetRemotePCMBlockStatus也可以用来检查闭塞解除对端电路群的操作是否完成。

注意事项：

- 本函数只支持 TUP 协议和 ISUP 协议；
- 按照 TUP 协议和 ISUP 协议的规定，电路群解除闭塞消息需要发送 2 次才有效，所以驱动程序会向对端交换机连续发送 2 次电路群解除闭塞消息，因此，应用程序只需调用本函数一次。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockRemotePCM，SsmGetRemotePCMBlockStatus

相关事件：E_CHG_RemotePCMBlock

2.20.1.2.2.3 SsmGetRemotePCMBlockStatus

获取电路群的闭塞状态。

函数原型：

int SsmGetRemotePCMBlockStatus(int nLocalPcmNo, DWORD dwBlockMode)

参数说明：

nLocalPcmNo	PCM 逻辑编号

dwBlockMode	选择希望查询的电路群闭塞类型。取值范围为： TUP 协议： 1：面向维护的电路群闭塞状态； 2：面向硬件故障的电路群闭塞状态； 3：软件产生的电路群闭塞状态； ISUP 协议： 0：面向维护的电路群闭塞状态； 1：面向硬件故障的电路群闭塞状态；

返回值：

-1	调用失败
0	没有闭塞对端电路群，或者已经成功解除闭塞对端电路群
1	已经成功闭塞对端电路群
2	已经发出电路群闭塞消息，正在等待对端交换机的证实信号
3	已经发出电路群解除闭塞消息，正在等待对端交换机的证实信号

功能描述：

获取电路群的闭塞状态。返回结果可以用来检查函数 SsmBlockRemotePCM 或 SsmUnblockRemotePCM 的执行情况。

注意事项：

- 本函数只支持 TUP 协议和 ISUP 协议。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBlockRemotePCM，SsmUnblockRemotePCM

2.20.2 基于 SS7-MTP3 的高级编程 API

2.20.2.1 SsmSendSs7Msu

向对端信令点发送一条消息（MSU）。

函数原型：

int SsmSendSs7Msu(WORD wMsuLength, PUCHAR pucMsuBuf)

参数说明：

wMsuLength	发送 MSU 的长度，单位为字节，最大不能超过 273
pucMsuBuf	存放 MSU 数据的首地址指针。pucMsuBuf 中的第 1 个字节（即 pucMsuBuf[0]）必须为 SIO 字段。

返回值：

0	发送成功
-1	发送失败。失败原因可能为： ✧ MTP3 业务中断 ✧ 系统不支持 SS7 信令 ✧ 系统的 API 接口函数尚未开放

功能描述：

向对端信令点发送一条消息（MSU）。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetSs7Msu

2.20.2.2SsmGetSs7Msu

查询驱动程序的 MSU 接收缓冲区中是否有 MSU 消息。

函数原型：

int SsmGetSs7Msu(PUCHAR* ppucMsuBuf)

参数说明：

ppucMsuBuf	返回保存在驱动程序内部缓冲区的 MSU 消息的首地址指针。MSU 消息的第 1 个字节为 SIO 字段。（注意：ppucMsgBuf 指针应该在调用前声明，并且不能指定其长度）

返回值：

0	驱动程序内部缓冲区的 MSU 消息接收缓冲区为空
>0	当前 MSU 的长度，单位为字节
-1	操作失败。失败原因可能是： ✧ 系统不支持 SS7 信令 ✧ 系统的 API 接口函数尚未开放

功能描述：

查询驱动程序中的 MSU 接收缓冲区中是否有 MSU 消息，如果有，取出最早收到的消息。

每当驱动程序收到一条 MSU，会将其保存到驱动程序内部的 MSU 接收缓冲区中，然后向应用程序抛出 E_RCV_Ss7Msu 消息。

注意事项：

- 驱动程序内部的 MSU 接收缓冲区填满后，后续到达的 MSU 将被丢弃。因此，应用程序应保证尽快将消息取出并处理。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendSs7Msu

相关事件：E_RCV_Ss7Msu

2.20.2.3SsmGetMtp3State

2.20.2.4SsmGetMtp3StateEx

获取源信令点到目的地信令点之间的业务是否开通。

函数原型：

int SsmGetMtp3State()

int SsmGetMtp3StateEx(int nDpcNo)

参数说明：

nDpcNo	目的地信令点的编号。目的地的信令点编号与信令点编码之间的对应关系在 SS7 信服务器的配置程序中指定。

返回值：

1	业务开通
0	业务中断
-1	调用失败

功能描述：

获取源信令点到目的地信令点之间的业务是否开通。只有当源信令点到目的地信令点处于业务开通状态时，应用程序才能调用函数 SsmSendSs7Msu 向 DPC 发送消息。

SsmGetMtp3StateEx 支持多个目的地信令点的应用场合，而 SsmGetMtp3State 是一个老版本的函数，只适用于系统中与单个目的地信令点直连的应用场合，相当于将 nDpcNo 设置为 0 时对 SsmGetMtp3StateEx 进行调用。

注意事项：

- 由于 SS7 信令具有自恢复功能，因此当应用程序使用本函数作为 SS7 信令系统监控的告警源时，应该在检测到业务中断并持续一定时间后，才发出告警信号；
- 当 SS7 Server 检测到到达目的地信令点的链路状态发生变化时，在保存该信令点的业务状态后，会向驱动程序抛出 E_CHG_Mtp3State 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmGetMtp2Status

相关事件：E_CHG_Mtp3State

2.20.2.5 SsmGetMtp2Status

获取 SS7 信令链路组中的 64kbps 信令链路的工作状态。

函数原型：

int SsmGetMtp2Status(int nLinkNum)

参数说明：

nLinkNum	信令链路编号，在 SS7 信服务器的配置程序中指定

返回值：

0	DSP 的程序已经上载成功，但未能启动运行
1	业务中断
2	初始定位中
3	已定位/准备好
4	已定位/未准备好
5	业务开通
6	处理机故障
-1	调用失败

功能描述：

本函数取得 SS7 信令下信令链路的状态，只有在状态等于 5（业务开通）时，该信令链路才可用。

注意事项：

- 本函数返回 MTP2 层的信息，通常用于显示等监视和维护目的。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMtp3State，SsmGetMtp3StateEx

2.20.2.6 SsmSendSs7MsuEx

根据输入的通道号（ch）对消息头打包，消息内容由应用程序自行配置。

函数原型：

int WINAPI SsmSendSs7MsuEx(int ch, int nNewStep, WORD wMsuLength, PUCHAR pucMsuBuf)

参数说明：

ch	通道号
nNewStep	预留
wMsuLength	消息内容长度，内容长度从消息类型开始计算，不包括消息头
pucMsuBuf	消息长度

返回值：

-1	失败，参数错误
0	失败，发送失败
1	成功

功能描述：

该函数根据输入的通道号（ch）对消息头打包，消息内容由应用程序自行配置。

2.20.3 有关 ISDN 信令的高级函数

2.20.3.1 SsmGetIsdnMsu

从驱动程序的内部缓冲区中取出一条 ISDN 消息。

函数原型：

int SsmGetIsdnMsu(int nPcmId, PUCHAR pucMsuBuf)

参数说明：

nPcmId	数字中继线的逻辑编号
pucMsuBuf	返回 ISDN 消息。pucMsuBuf 由应用程序分配存储空间，大小不能低于 256 字节

返回值：

0	缓冲区中没有消息
>0	pucMsuBuf 中消息的实际长度，单位为字节。
-1	调用失败

功能描述：

从驱动程序的内部缓冲区中取出一条 ISDN 消息。

注意事项：

- 只有当配置项AutoHandleIsdn设置为 1 时，驱动程序才会将收到的ISDN消息保存到内部缓冲区中。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmCheckIsdnMsu

2.20.3.2 SsmSendIsdnMsu

发送一条 ISDN 消息。

函数原型：

int SsmSendIsdnMsu(int nPcmId, int nMsgLen, PUCHAR pucMsuBuf)

参数说明：

nPcmId	数字中继线的逻辑编号
nMsgLen	pucMsuBuf 中消息的实际长度，单位为字节
pucMsuBuf	ISDN 消息缓冲区

返回值：

0	调用成功
-1	调用失败

功能描述：

在指定的数字中继线的 16 时隙上发送一条 ISDN 消息。

注意事项：

- 只有当配置项AutoHandleIsdn设置为 1 时，才可以调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIsdnMsu

2.20.3.3 SsmCheckIsdnMsu

查询驱动程序的内部缓冲区中是否有 ISDN 消息。

函数原型：

int SsmCheckIsdnMsu(int nPcmId)

参数说明：

nPcmId	数字中继线的逻辑编号

返回值：

-1	错误
≥0	ISDN 消息的条数

功能描述：

查询驱动程序的内部缓冲区中是否有 ISDN 消息。

注意事项：

- 只有当配置项AutoHandleIsdn设置为 0 时，调用本函数才有意义。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetIsdnMsu

2.20.3.4 SsmISDNGetStatus

获取 ISDN 协议 HDLC 链路的状态。

函数原型：

int SsmISDNGetStatus(int nPcmNo, int *pL3Start, int *pL2DStatus, int *pL2D_L3Atom, int * pL3_L2DAtom, int *pRef_ind)

参数说明：

nPcmNo	本地数字中继线的逻辑编号
pL3Start	返回 ISDN 协议第 3 层（L3）是否启动的信息。返回值为 1 表示启动，0 表示未启动。
pL2Dstatus	返回 L3 的工作状态
pL2D_L3Atom	返回接收到的第 2 层（L2）到 L3 的消息原语
pRef_ind	返回接收到标示值

返回值：

0	调用成功
-1	调用失败

功能描述：

获取 ISDN 协议 HDLC 链路的状态。

注意事项：

- 本函数通常在 E_CHG_ISDNStatus 事件的响应代码中调用。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：E_CHG_ISDNStatus

2.20.4 有关中国 1 号信令的高级函数☑

2.20.4.1 控制 ABCD 信令比特

2.20.4.1.1SsmGetCAS

获取通道上 ABCD 信令的接收端状态。

函数原型：

int SsmGetCAS(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	返回值的高 4 位比特为 0，最低 4 个比特为本端收到的 CAS 信令比特： bit3 bit2 bit1 bit0 A B C D

功能描述：

获取对端交换机发送的 ABCD 信令的当前值。

注意事项：

- ABCD 信令在 E1 的 16 时隙中传递；
- 本函数实现的功能也可以通过 E_RCV_CAS 事件获取，推荐使用 E_RCV_CAS 事件方式。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendCAS

2.20.4.1.2SsmSendCAS

设置通道上 ABCD 信令的发送端状态。

函数原型：

int SsmSendCAS(int ch, BYTE btCas)

参数说明：

ch	通道编号
btCas	发送 CAS 信令值，高 4 位比特为 0，低 4 位比特为 ABCD 信令比特的新状态： bit3 bit2 bit1 bit0 A B C D

返回值：

-1	调用失败
0	调用成功

功能描述：

设置本端 ABCD 信令的状态。

注意事项：

- ABCD 信令在 E1 的 16 时隙中传递；
- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetSendingCAS，SsmGetCAS

2.20.4.1.3SsmGetSendingCAS

获取通道上 ABCD 信令的发送端状态。

函数原型：

int SsmGetSendingCAS(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
≥0	返回值的最低 4 个比特为发送端的 ABCD 信令比特： bit3 bit2 bit1 bit0 A B C D 其余比特保留。

功能描述：

获取通道上 ABCD 信令的发送端状态。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetCAS，SsmSendCAS

2.20.4.1.4SsmSetSendCASFlag

设置通道是否允许发送 CAS 信号。

函数原型：

int WINAPI SsmSetSendCASFlag(int ch, int nCASFlag)

参数说明：

ch	通道编号
nCASFlag	是否允许发送 CAS 信号 0：不允许 1：允许

返回值：

-1	调用失败
=0	调用成功

功能描述：

设置通道是否允许发送 CAS 信号。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetSendCASFlag，SsmSendCAS

2.20.4.1.5SsmGetSendCASFlag

获取通道是否允许发送 CAS 信号的标志。

函数原型：

int WINAPI SsmGetSendCASFlag(int ch, int* pCASFlag)

参数说明：

ch	通道编号
pCASFlag	得到是否允许发送 CAS 信号标志的指针

返回值：

-1	调用失败
=0	调用成功

功能描述：

获取通道是否允许发送 CAS 信号的标志。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetCAS，SsmSetSendCASFlag

2.20.4.2 控制 R2 信号收发器

2.20.4.2.1 SsmSetRxR2Mode

设置 R2 信号接收器。

函数原型：

int SsmSetRxR2Mode(int ch, int nMode, BOOL bEnable)

参数说明：

ch	通道编号
nMode	R2 信号接收器的工作模式： =0：接收后向 R2 信号 =1：接收前向 R2 信号
bEnable	R2 信号接收器的工作状态： =TRUE： 开启 =FALSE： 关闭

返回值：

-1	调用失败
0	调用成功

功能描述：

设置 R2 信号接收器。

注意事项：

- 如果通道配置为自动接续，调用本函数时将得到返回值-1；
- R2 信号收发器和DTMF检测器不能同时工作，因此，使用R2 信号收发器时，必须关闭DTMF检测器。MFC过程结束后，需要自行调用SsmEnableRxDtmf启动DTMF检测器。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相关函数：SsmGetR2

2.20.4.2.2 SsmGetR2

获取 R2 信号接收器的检测结果。

函数原型：

int SsmGetR2(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	线路上没有检测到 R2 信号
1~15	线路上检测到了 R2 信号。对于前向 R2 信号，返回值范围为 1~15；对于后向 R2 信号，返回值范围为 1~6

功能描述：

获取 R2 信号接收器的检测结果。

注意事项：

- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetRxR2Mode，SsmSendR2，SsmSendR2Ex

2.20.4.2.3 SsmSendR2

在通道上产生一个 R2 信号。

函数原型：

int SsmSendR2(int ch, int nMode, BYTE btR2)

参数说明：

ch	通道编号
nMode	R2 信号发生器的工作模式： =0：产生后向 R2 信号 =1：产生前向 R2 信号
btR2	R2 信号的编码值： ● 对于前向 R2 信号，取值范围为：1~15 ● 对于后向 R2 信号，取值范围为：1~6

返回值：

-1	调用失败
0	调用成功

功能描述：

在通道上产生一个R2 信号，并一直持续到应用程序调用SsmStopSendR2函数后才会停止。

注意事项：

- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopSendR2，SsmGetSendingR2，SsmGetR2，SsmSendR2Ex

2.20.4.2.4SsmSendR2Ex

在通道上产生一个指定时间长度的 R2 信号。

函数原型：

int SsmSendR2Ex(int ch, int nMode, BYTE btR2, DWORD dwKeepTime)

参数说明：

ch	通道编号
nMode	R2 信号发生器的工作模式： =0：产生后向 R2 信号 =1：产生前向 R2 信号
btR2	R2 信号的编码值： ● 对于前向 R2 信号，取值范围为：1~15 ● 对于后向 R2 信号，取值范围为：1~6
dwKeepTime	R2 信号保持时间，单位为毫秒

返回值：

-1	调用失败
0	调用成功

功能描述：

在通道上产生一个持续时间为dwKeepTime 毫秒的R2 信号，应用程序也可以调用SsmStopSendR2函数停止R2信号。

注意事项：

- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmStopSendR2，SsmGetSendingR2，SsmGetR2，SsmSendR2

2.20.4.2.5SsmStopSendR2

停止在通道上产生一个 R2 信号。

函数原型：

int SsmStopSendR2(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

停止在通道上产生一个 R2 信号。

注意事项：

- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendR2，SsmSendR2Ex

2.20.4.2.6 SsmGetSendingR2

获取通道上 R2 信号发生器的工作模式。

函数原型：

int SsmGetSendingR2(int ch, int* pnMode, BYTE* pbtR2)

参数说明：

ch	通道编号
pnMode	返回 R2 信号发生器的工作模式： =0：产生后向 R2 信号 =1：产生前向 R2 信号
pbtR2	返回 R2 信号的编码值： 对于前向 R2 信号，取值范围为：1~15 对于后向 R2 信号，取值范围为：1~6

返回值：

-1	调用失败
0	调用成功

功能描述：

获取通道上 R2 信号发生器的工作模式。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSendR2，SsmSendR2Ex

2.21 SHV 系列板卡的专用函数（CTI 系列）

2.21.1 SsmGetMaxVCh

获取应用系统内变声资源通道的总数。

函数原型：

int SsmGetMaxVCh()

参数说明：无

返回值：

-1	调用失败
≥0	变声资源通道的总数

功能描述：

获取应用系统内变声资源通道的总数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMaxFreeVCh

2.21.2 SsmGetMaxFreeVCh

获取空闲的变声资源通道的总数。

函数原型：

int SsmGetMaxFreeVCh()

参数说明：无

返回值：

-1	调用失败
≥0	空闲的变声资源通道的总数

功能描述：

获取空闲的变声资源通道的总数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMaxVCh

2.21.3 SsmBindVCh

将语音通道绑定到变声资源通道。

函数原型：

int SsmBindVCh(int iCh)

参数说明：无

iCh	语音通道编号

返回值：

0	调用成功
-1	调用失败：iCh 越界
-2	调用失败：iCh 不支持总线交换功能
-3	调用失败：没有空闲的变声资源通道
-4	调用失败：iCh 已经绑定到某个变声资源通道
-5	调用失败：驱动程序尚未加载

功能描述：

将语音通道绑定到变声资源通道。

驱动程序执行本函数时，会自动分配一个空闲的变声资源通道，并将此变声资源通道绑定到语音通道 iCh 上。iCh 通道的来话信号会首先进入变声资源通道进行变声处理，然后再送到交换总线上。

注意事项：

- 驱动程序加载成功后，应用程序可以在任何时候调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmUnBindVCh，SsmSetVoiceEffect

代码范例：

例 1：通道 1 与通道 2 进行双向连接，通道 1 的声音需要变声，通道 2 的声音不需要变声，实现代码如下：

```
……                          //发起呼叫的代码
SsmBindVCh(1);              //将通道 1 绑定到一个变声资源通道
SsmSetVoiceEffect(1,180);   //设置变声效果
SsmTalkWith(1,2);           //建立通道 1 与通道 2 的双向连接
……                          //通话
SsmUnBindVCh(1);            //通话结束，解除通道 1 与通道 2 的双向连接
SsmStopTalkWith(1,2);       //拆除双向连接
```

例 2：通道 1、通道 2 和通道 3 加入一个会议室，通道 1、通道 2 要求变声功能。实现代码如下：

```
int GroupID = SsmCreateConfGroup(…);    //创建一个会议室
……
SsmBindVCh(1);
SsmJoinConfGroup(GroupID,1,…);          //通道 1 加入会议室
SsmBindVCh(2);
SsmJoinConfGroup(GroupID,2,…);          //通道 2 加入会议室
SsmJoinConfGroup(GroupID,3,…);          //通道 3 加入会议室
……
```

2.21.4 SsmUnBindVCh

解除语音通道与变声资源通道的绑定。

函数原型：

int SsmUnBindVCh(int iCh)

参数说明：

iCh	语音通道编号

返回值：

0	调用成功
-1	调用失败：iCh 越界
-2	调用失败：iCh 不支持总线交换功能
-3	调用失败：iCh 没有绑定到某个变声资源通道
-4	调用失败：驱动程序尚未加载

功能描述：

解除语音通道与变声资源通道的绑定。变声资源通道解除绑定后，会由驱动程序自动回收，进入空闲队列。

注意事项：

- 如果语音通道不再需要变声功能，应解除与变声资源通道的绑定。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBindVCh

2.21.5 SsmSetVoiceEffect

设置变声的效果。

函数原型：

int SsmSetVoiceEffect(int iCh, int iValue)

参数说明：

iCh	语音通道编号
iValue	变声的效果，取值范围：70~220，128 表示原始声音（即不变声），大于 128 为女声效果，小于 128 为男生效果。

返回值：

0	调用成功
-1	调用失败：iCh 越界
-2	调用失败：iCh 没有绑定到某个变声资源通道
-3	调用失败：iValue 越界
-4	调用失败：驱动程序尚未加载

功能描述：

设置变声的效果。

注意事项：

- 本函数必须在语音通道已经绑定到某个变声资源通道后才能调用；
- 变声的效果难以具体描述，请根据实际测试的效果对 iValue 进行设置。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmBindVCh，SsmGetVoiceEffect

2.21.6 SsmGetVoiceEffect

获取变声的效果。

函数原型：

int SsmGetVoiceEffect(int iCh)

参数说明：

iCh	语音通道编号

返回值：

>0	调用成功，返回变声的效果，具体取值请参见函数SsmSetVoiceEffect的说明。
-1	调用失败：iCh 越界
-2	调用失败：iCh 没有绑定到某个变声资源通道
-3	调用失败：驱动程序尚未加载

功能描述：

获取变声的效果。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetVoiceEffect

2.22 SHN 系列板卡的专用函数（CTI 系列）

2.22.1 SIP 通道的专用编程接口

2.22.1.1 SsmIpGetUsingCodecType

取得指定通道当前通话中实际使用的 Codec 类型。

函数原型：

int SsmGetIpUsingCodecType (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
6	指定通道当前正在使用 A-Law 格式进行通话
7	指定通道当前正在使用 μ-Law 格式进行通话
131	指定通道当前正在使用 G.729A 格式进行通话

功能描述：

取得指定通道当前通话中实际使用的 Codec 类型。

注意事项：

- 只有在通道处于通话状态时调用本函数才有意义。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.2 SsmIpSetForwardNum

设置指定通道的转寄号码。

函数原型：

int SsmIpSetForwardNum(int ch, LPSTR pszForwardNum)

参数说明：

ch	通道编号
pszForwardNum	转寄号码

返回值：

-1	调用失败
0	调用成功

功能描述：

设置指定通道的转寄号码。设置以后，对该通道的任何呼入将无条件地转寄到设置的号码。

注意事项：无。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.3 SsmIpInitiateTransfer

设置指定通道的转移号码。

函数原型：

int SsmIpInitiateTransfer(int ch, LPSTR pszTransferTo)

参数说明：

ch	通道编号
pszTransferTo	转移号码

返回值：

-1	调用失败
0	调用成功

功能描述：

设置指定通道的转移号码。对通话中的通道调用该函数，将该次呼叫转移到设置的号码上。

注意事项：

- 只有在通道处于通话状态时调用本函数才有意义。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.4 SsmIpGetMessageField

取得指定通道上用于建立呼叫的相关消息中的一系列字段值。

函数原型：

int SsmIpGetMessageField(int ch, int type, LPSTR szBuffer, int *pSize)

参数说明：

ch	通道编号
Type	需要取得的字段类型
SzBuffer	用于保存取得的字段内容的缓冲区
PSize	该指针指向的值为用户提供的缓冲区大小。当本函数返回后，如果成功，该指针指向的值为取得的有效数据大小；如果失败，该指针指向的值为实际需要的缓冲区大小。

返回值：

-1	调用失败
0	调用成功

功能描述：

取得指定通道上用于建立呼叫的相关消息中的一系列字段值。

参数 type 的取值如下：

- TYPE_MESSAGE_CONTACT

  取得 INVITE 或者 REINVITE 消息中的 Contact 字段内容。该参数仅在作为被叫方使用时有效。如果消息体中存在多个 Contact 字段，那么返回的格式如下：

Contact0	\0	Contact1	\0	…	Contactn	\0	\0

注意事项：

- 只有在通道处于通话状态时调用本函数才有意义。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2 H.323 通道的专用编程接口

2.22.2.1 SsmIpGetUsingCodecType

取得指定通道当前通话中实际使用的 Codec 类型。

函数原型：

int SsmIpGetUsingCodecType (int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
6	指定通道当前正在使用 A-Law 格式进行通话
7	指定通道当前正在使用 μ-Law 格式进行通话
131	指定通道当前正在使用 G.729A 格式进行通话

功能描述：

取得指定通道当前通话中实际使用的 Codec 类型。

注意事项：

- 只有在通道处于通话状态时调用本函数才有意义。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.23 ATP 系列录音卡专用函数（REC 系列）

2.23.1 麦克风通道的输入增益控制

2.23.1.1 SsmQueryOpMicGain

查询通道是否支持设置输入信号增益的操作。

函数原型：

int SsmQueryOpMicGain(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	不支持
1	支持

功能描述：

查询通道是否支持设置输入信号增益的操作。

注意事项：

- 只有 ATP 系列板卡的模拟录音通道（包括模拟中继线通道和麦克风录音通道）才支持此项功能。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetMicGain

2.23.1.2 SsmSetMicGain

设置模拟录音通道的输入信号的增益。

函数原型：

int SsmSetMicGain(int ch, int nMicGain)

参数说明：

ch	通道编号
nGain	增益值： 0：正常增益 1：提升 20DB

返回值：

-1	调用失败
0	设置成功

功能描述：

设置模拟录音通道的输入信号的增益。

注意事项：

- 配置项MicGain可以实现相同功能；
- 本函数只适用于 ATP 系列板卡的模拟录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmGetMicGain

2.23.1.3 SsmGetMicGain

获取模拟录音通道上设置的输入信号增益值。

函数原型：

int SsmGetMicGain(int ch)

参数说明：

ch	通道编号

返回值：

-1	调用失败
0	正常增益
1	高增益（20DB）

功能描述：

获取模拟录音通道上设置的输入信号增益值。

注意事项：

- 本函数只适用于 ATP 系列板卡的模拟录音通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmSetMicGain

2.24 DTP 系列板卡的专用函数（REC 系列）

2.24.1 基于状态机编程模式的函数

2.24.1.1 获取基本信息的函数

2.24.1.1.1 SpyGetMaxCic

获取系统中被监控电路的总数。

函数原型：

int SpyGetMaxCic()

参数说明：无

返回值：

-1	调用失败
≥0	SpyCic 的总数

功能描述：

获取系统中被监控电路的总数，即 [AppSpyCICTable]配置节中的TotalAppSpyCIC配置项的设定值。有关SpyCic的更多信息，请参见第 1 章中“DTP系列板卡”部分内容。

注意事项：

- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SpyChToCic

2.24.1.1.2 SpyChToCic

根据通道的逻辑编号来查找对应的 SpyCic 的逻辑编号。

函数原型：

int SpyChToCic(int ch)

参数说明：

ch	通道的逻辑编号

返回值：

返回监控电路状态

-1	失败
≥0	对应的 SpyCic 的逻辑编号

功能描述：

根据通道的逻辑编号来查找对应的SpyCic的逻辑编号。有关SpyCic的逻辑编号的更多信息，请参见第 1 章中“DTP系列板卡”部分内容。

注意事项：

- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SpyGetMaxCic

2.24.1.2 获取呼叫进展的函数

2.24.1.2.1 SpyGetState

获取驱动程序基于 SpyCic 的状态机的当前状态。

函数原型：

int SpyGetState(int nCic)

参数说明：

nCic	被监控电路编号

返回值：

返回值为-1 表示调用失败，其它返回值的含义如下表所示：

值	shpa3api.h 中的宏	状态描述
0	S_SPY_STANDBY	空闲
2	S_SPY_RINGING	振铃
3	S_SPY_TALKING	连接（通话）。当 SpyCic 迁移到进入 S_SPY_TALKING 状态后，应用程序可以： ✧ 调用SpyRecToFile启动录音； ✧ 调用SpyGetCalleeId取得被叫号码； ✧ 调用SpyGetCallerId取得主叫号码； ✧ 以主叫方通道或被叫方通道为参数，调用基于通道的函数获取其它相关的信息。
105	S_SPY_RCVPHONUM	接收被叫号码
110	S_SPY_SS1RESET	电路复原
111	S_SPY_SS1WAITBWDACK	SS1：等待后向证实
112	S_SPY_SS1WAITKB	SS1：等待 KB 信号
其它	保留	

功能描述：

获取驱动程序基于 SpyCic 的状态机的当前状态。

注意事项：

- 建议使用 E_CHG_SpyState 事件来替代本函数；
- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SpyGetCalleeId，SpyGetCallerId，SpyGetCallOutCh，SpyGetCallInCh

2.24.1.3获取呼叫方向的函数

2.24.1.3.1SpyGetCallInCh

2.24.1.3.2SpyGetCallOutCh

获取当前呼叫中的主叫方通道（SpyGetCallOutCh）或被叫方通道（SpyGetCallInCh）的逻辑编号。

函数原型：

int SpyGetCallInCh(int nCic)

int SpyGetCallOutCh(int nCic)

参数说明：

nCic	SpyCic的逻辑编号。有关SpyCic的更多信息参见第 1 章中“DTP系列板卡”部分内容。

返回值：

-1	调用失败
≥0	SpyGetCallInCh 函数： 返回被叫方通道的逻辑编号 SpyGetCallOutCh 函数：返回主叫方通道的逻辑编号

功能描述：

获取当前呼叫中的主叫方通道（SpyGetCallOutCh）或被叫方通道（SpyGetCallInCh）的逻辑编号。有关被叫方通道的更多信息请参见第 1 章中“DTP系列板卡”部分内容。

注意事项：

当 SpyCic 的状态迁移到 S_SPY_RINGING 或 S_SPY_TALKING 时，调用 SpyGetCallOutCh 或 SpyGetCallInCh 才能得到有效信息；

本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SpyGetCallerId，SpyGetCalleeId

2.24.1.4 取得主叫方号码和被叫方号码的函数

2.24.1.4.1SpyGetCallerId

2.24.1.4.2SpyGetCalleeId

获取本次呼叫的主叫方号码（SpyGetCallerId）或被叫方号码（SpyGetCalleeId）。

函数原型：

int SpyGetCallerId(int nCic, char *pszCallingPartyNumber)

int SpyGetCalleeId(int nCic, char * pszCalledPartyNumber)

参数说明：

nCic	SpyCic 的逻辑编号

pszCallingPartyNumber	存放主叫方号码的缓冲区指针，由应用程序分配存储空间，不能少于 50 个字符。
pszCalledPartyNumber	存放被叫方号码的缓冲区指针，由应用程序分配存储空间，，不能少于 50 个字符。

返回值：

-1	调用失败
≥0	号码的长度（个数）

功能描述：

呼叫建立后，SpyGetCallerId 函数获取本次呼叫的主叫方号码，SpyGetCalleeId 获取本次呼叫的被叫方号码。

注意事项：

- 当 SpyCic 的状态迁移到 S_SPY_RINGING 或 S_SPY_TALKING 时，调用 SpyGetCallerId 或 SpyGetCalleeId 才能得到完整的信息；
- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SpyGetCallOutCh，SpyGetCallInCh

2.24.1.5 获取数字中继线状态的函数

2.24.1.5.1SpyGetLinkstatus

获取被监控的数字中继线的工作状态。

函数原型：

int SpyGetLinkStatus(int nSpyPcmNo,UCHAR ucFlag);

参数说明：

nSpyPcmNo	被监控PCM的逻辑编号，取值范围：由配置项TotalSpyPcm决定。
ucFlag	0：选择上行数字中继线 1：选择下行数字中继线

返回值：

-1	失败
0	故障
1	正常

功能描述：

获取被监控的数字中继线的工作状态。

注意事项：

- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.24.1.6 获取 SS7 信令呼叫状态机同步输出的原始信令消息

2.24.1.6.1 SsmGetSs7SpyMsu

获取 SS7 信令呼叫状态机同步输出的原始信令消息。

函数原型：

int SsmGetSs7SpyMsu (PUCHAR* ppucMsuBuf)

参数说明：

ppucMsuBuf	保存原始的 ISUP/TUP 消息的缓冲区指针，由应用程序分配存储空间，不能少于 273 个字节。

返回值：

0	没有消息
>0	返回消息的长度，单位为字节
-1	调用失败。

功能描述：

获取 SS7 信令呼叫状态机同步输出的原始信令消息。取出消息的方式为“先进先出”，即总是返回最先收到的消息。

注意事项：

- 只有当配置项bOpenSpySS7Msu设置为 1 时，驱动程序才会将收到的ISUP/TUP消息复制到内部缓冲区，本函数才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：无

2.25 DST 系列录音卡的专用函数（REC 系列）

2.25.1 基于状态机编程模式的函数

2.25.1.1 DTRGetLCDStr

取得 LCD 显示屏的当前的显示信息

函数原型：

int DTRGetLCDStr(int ch, LPSTR pszStr)

参数说明：

ch	通道号
pszStr	存放 LCD 信息的字符串指针，由应用程序分配存储空间，空间应该大于或者等于 255 个字节

返回值：

-1	调用失败
0	表示 LCD 显示信息为空
>0	LCD 显示信息的长度

功能描述：

取得被录音的数字电话机的 LCD 当前的显示信息，多数情况下，该信息和话机 LCD 显示屏上正在显示的信息相同。某些情况下，数字电话机自主更新显示内容，这些内容并不在电话线上传送，因此本函数并不能在所有的交换机和话机环境下都能正确工作，并且所获取的信息的实际含义根据不同的交换机和电话机会有所不同，具体情况应在实际环境下调试确定。

注意事项：

- 本函数只支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.25.1.2 DTRGetDKeyStr

取得话机数字按键操作

函数原型：

int DTRGetDKeyStr(int ch, LPSTR pszDKeyStr)

参数说明：

ch	通道号
pszDKeyStr	存放 DKey 信息的字符串指针，由应用程序分配存储空间，空间应该大于或者等于 255 个字节

返回值：

-1	调用失败
0	表示 DKey 信息为空
>0	DKey 信息的长度

功能描述：

取得录音数字电话在“Active”状态的数字按键信息，每一个新的数字按键都会被添加到缓冲区的尾部，该缓冲在语音通道关闭时（“Active”转移到“空闲”）被驱动清空。该函数只能取得对“0-9、*、#”这 12 个按键的操作。某些数字话机的号码按键并不产生 D 信道事件，而是和模拟电话机一样直接发送 DTMF 信号，这种情况下，本函数不能取得这些按键操作。

注意事项：

- 本函数只支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：DTRGetLCDStrA

2.25.2 高级函数

2.25.2.1 DTRGetLCDStrA

取得 LCD 显示屏的当前的显示信息

函数原型：

char* DTRGetLCDStrA(int ch)

参数说明：

ch	通道号

返回值：

NULL	表示 LCD 显示信息为空
其他	存放 LCD 显示信息字符串的指针

功能描述：

取得被录音的数字电话机的 LCD 当前的显示信息。多数情况下，该信息和话机 LCD 显示屏上正在显示的信息相同。某些情况下，数字电话机自主更新显示内容，这些内容并不在电话线上传送，因此本函数并不能在所有的交换机和话机环境下都能正确工作，并且所获取的信息的实际含义根据不同的交换机和电话机会有所不同，具体情况应在实际环境下调试确定。本函数专用于在 PowerBuilder 编程时取代 DTRGetLCDStr 函数。

注意事项：

- 本函数只支持 DST 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：DTRGetLCDStr

2.26 有关语音编码格式转换的函数

2.26.1 fPcm_Mp3ConvertALaw

2.26.2 fPcm_Mp3ConvertULaw

fPcm_Mp3ConvertALaw 将 Synway 板卡录制的 IMP3 格式文件转换为 A 率格式文件，fPcm_Mp3ConvertULaw 将 MP3 格式文件转换为μ率格式文件。

函数原型：

int fPcm_Mp3ConvertALaw (char*szSourecFile, char*szTargetFile)

int fPcm_Mp3ConvertULaw (char*szSourecFile, char*szTargetFile)

参数说明：

szSoureFile	使用 Synway 板卡录制的 MP3 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件。
SzTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

fPcm_Mp3ConvertALaw 将 MP3 格式文件转换为 A 率格式文件，fPcm_Mp3ConvertULaw 将 MP3 格式文件转

换为μ率格式文件。源文件必须是采用 Synway 板卡录制的 MP3 无格式文件或标准 wav 文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用；
- 本函数执行时，需要用到 MP3 的 ACM 解码引擎。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.26.3 fPcm_AdpcmToAlaw

2.26.4 fPcm_AdpcmToGsm

2.26.5 fPcm_AdpcmToMp3

将 Synway 板卡录制的 IMA ADPCM 格式文件转换为 A 率格式文件（fPcm_AdpcmToAlaw）或 GSM 6.10 格式文件（fPcm_AdpcmToGsm），或 MP3 格式文件（fPcm_AdpcmToMp3）。

函数原型：

int fPcm_AdpcmToAlaw(char*szSoureFile, char* szTargetFile)

int fPcm_AdpcmToGsm(char*szSoureFile,char*szTargetFile)

int fPcm_AdpcmToMp3(char*szSoureFile, char* szTargetFile)

参数说明：

szSoureFile	用 Synway 板卡录制的 IMA ADPCM 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

fPcm_AdpcmToAlaw 将 Synway 板卡录制的 IMA ADPCM 格式文件转换为 A 率格式文件，fPcm_AdpcmToGsm 将 IMA ADPCM 格式文件转换为 GSM 6.10 格式文件，fPcm_AdpcmToMp3 将 IMA ADPCM 格式文件转换为 Mp3 格式文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	fPcm_AdpcmToAlaw 要求 SynCTI Ver. 4.0 或更高； fPcm_AdpcmToGsm 要求 SynCTI Ver. 4.7.2.0 或更高； fPcm_AdpcmToMp3 要求 SynCTI Ver. 4.7.3.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_MemAdpcmToALAW，fPcm_MemAdpcmToULAW

2.26.6 fPcm_ AlawToUlaw

2.26.7 fPcm_ UlawToAlaw

将 Synway 板卡录制的 A-Law 格式文件转换为 μ-Law 格式文件（fPcm_ AlawToUlaw）或者将 μ-Law 格式文件转换为 A-Law 格式文件（fPcm_ UlawToAlaw）。

函数原型：

int WINAPI fPcm_AlawToUlaw(char*szSourceFile, char*szTargetFile)

int WINAPI fPcm_UlawToAlaw(char*szSourceFile, char*szTargetFile)

参数说明：

szSoureFile	用 Synway 板卡录制的 A-Law 或者 μ-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

fPcm_AlawToUlaw 将 Synway 板卡录制的 A-Law 格式文件转换为 μ-Law 格式文件，fPcm_UlawToAlaw 将 μ-Law 格式文件转换为 A-Law 格式文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.26.8 fPcm_MemAdpcmToALAW

2.26.9 fPcm_MemAdpcmToULAW

将 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 A-law 或μ-law。fPcm_MemAdpcmToALAW 将 IMA ADPCM 格式转换为 A-law，fPcm_MemAdpcmToULAW 将 IMA ADPCM 格式转换为μ-law。

函数原型：

DWORD fPcm_MemAdpcmToALAW(char * pSource, DWORD dwSourceSize, char * pTarget, DWORD dwTargetSize)

DWORD fPcm_MemAdpcmToULAW(char * pSource, DWORD dwSourceSize, char * pTarget, DWORD dwTargetSize)

参数说明：

pSource	输入缓冲区的首地址指针
dwSourceSize	输入缓存的大小，单位为字节，由于 IMA ADPCM 格式的帧长度为 256 字节，因此本参数必须是 256 字节的整数倍。
pTarget	存放转换后数据的缓冲区(输出缓冲区)首地址指针，由应用程序分配存储空间。
dwTargetSize	输出缓冲区的大小。由于每帧 IMA ADPCM 格式数据转换成 A-law 或μ-law 后，会产生 505 字节的数据，因此，输出缓冲区的大小应该不小于 dwSourceSize / 256 * 505 + 1

返回值：

0	转换失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。
>0	转换后的语音数据字节数

功能描述：

将 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 A-law 或μ-law 格式。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_AdpcmToAlaw

2.26.10 fPcm_AlawConvertPcm16

2.26.11 fPcm_ALawConvertPcm8

2.26.12 fPCM_AlawConvertGC8

2.26.13 fPcm_ALawConvertMp3

将 A-law 格式文件转换为 16 Bit PCM 格式文件（fPcm_AlawConvertPcm16）、8 Bit 无符号 PCM 格式文件（fPcm_ALawConvertPcm8）、G.729A 格式文件（fPCM_AlawConvertGC8）或 MP3 格式文件（fPcm_ALawConvertMp3）。

函数原型：

int fPcm_AlawConvertPcm16(char*fnALaw, char* szTargetFile)

int fPcm_AlawConvertPcm8(char* fnALaw, char* szTargetFile)

int fPCM_AlawConvertGC8(char * fnALaw,char*szTargetFile)

int fPcm_AlawConvertMp3(char*szSourceFile, char*szTargetFile)

参数说明：

fnALaw	A-law 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得。

功能描述：

fPcm_AlawConvertPcm16 将 A-law 格式文件转换为 16 Bit PCM 格式文件；fPcm_ALawConvertPcm8 将 A-law 格式文件转换为 8 Bit 无符号 PCM 格式文件；fPCM_AlawConvertGC8 将 A-law 格式文件转换为 G.729A 格式文件；fPcm_ALawConvertMp3 将 A-law 格式文件转换为 MP3 格式文件，但需安装 MP3 录音引擎才有效。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	fPcm_AlawConvertPcm16、fPcm_ALawConvertPcm8 要求 SynCTI Ver. 4.0 或更高，fPCM_AlawConvertGC8 要求 SynCTI Ver. 4.7.1.8 或更高，fPcm_ALawConvertMp3 要求 SynCTI Ver. 4.8.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_Pcm16ConvertAlaw、fPcm_GC8Convert、fPcm_MemAlawToPcm8、fPcm_MemAlawToPcm16

2.26.14 fPcm_ULawConvertMp3

将 μ-Law 格式文件转换为 MP3 格式文件。

函数原型：

Int WINAPI fPcm_ULawConvertMp3(char*szSourceFile, char*szTargetFile)

参数说明：

szSoureFile	用 Synway 板卡录制的 μ-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

将 Synway 板卡录制的 μ-Law 格式文件转换为 MP3 格式文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数： fPcm_ALawConvertMp3

2.26.15 fPcm_MemAlawToPcm8

2.26.16 fPcm_MemAlawToPcm16

将保存在缓冲区中的 A-law 格式数据转换成 8 Bit 无符号 PCM 格式（fPcm_MemAlawToPcm8）或 16 Bit PCM 格式（fPcm_MemAlawToPcm16）。

函数原型：

DWORD fPcm_MemAlawToPcm8 (char*pSrcBuf,DWORD dwSrcSize,char *pDstBuf,DWORD dwDstSize8)
DWORD fPcm_MemAlawToPcm16(char* pSrcBuf,DWORD dwSrcSize,char * pDstBuf,DWORD dwDstSize16)

参数说明：

pSrcBuf	源数据的缓冲区指针，存放 A 率格式的语音数据。
dwSrcSize	pSrcBuf 的大小，单位为字节。
pDstBuf	存放转换后数据的缓冲区指针，由应用程序分配存储空间。
dwDstSize8 dwDstSize16	pDstBuf 的大小，单位为字节。dwDstSize8 必须大于或等于 dwSrcSize，dwDstSize16 必须大于或等于 dwSrcSize 的 2 倍。

返回值：

0	转换失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。
>0	转换的字节数。

功能描述：

fPcm_MemAlawToPcm8 将 A-law 转换为 8 Bit 无符号 PCM 格式，fPcm_MemAlawToPcm16 将 A-law 转换为 16 Bit PCM 格式。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_AlawConvertPcm16，fPcm_ALawConvertPcm8，fPcm_MemPcm8ToAlaw

2.26.17 fPcm_MemGSMToPcm16

将保存在缓冲区中的 GSM 格式数据转换成 16 Bit PCM 格式。

函数原型：

DWORD WINAPI fPcm_MemGSMToPcm16(char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize)

参数说明：

pSource	输入缓存的指针，存放 GSM 格式的语音数据。
dwSourceSize	输入缓存的大小，单位为字节。大于等于 2600，小于 3250。最好为 65 的整数倍。
pTarget	目标缓存指针，存放 pcm16 格式的语音数据。
dwTargetSize	目标缓存大小，必须大于等于输入缓存大小的 5 倍

返回值：

0	转换失败。
>0	转换的字节数。

功能描述：

采用内存方式实现从 GSM 到 Pcm16 的格式转化。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_AlawConvertPcm16，fPcm_ALawConvertPcm8，fPcm_MemPcm8ToAlaw

2.26.18 fPcm_Pcm16ConvertAlaw

将 16 Bit PCM 格式文件转换为 A-law 格式的文件。

函数原型：

int fPcm_Pcm16ConvertAlaw (char*fnPCM16, char* szTargetFile)

参数说明：

fnPCM16	16 Bit PCM 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得。

功能描述：

将 16 Bit PCM 格式文件转换为 A-law 格式的文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.7 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.26.19 fPcm_MemPcm8ToAlaw

将保存在缓冲区中的 8 Bit 无符号 PCM 格式数据转换成 A-law 格式。

函数原型：

int fPcm_MemPcm8ToAlaw(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)

参数说明：

szSource	输入缓冲区的首地址指针，存放 PCM8 率格式的语音数据
nSourceLen	szSource 的大小，单位为字节
szTarget	存放转换后的数据的缓冲区指针，由应用程序分配存储空间
nTargetLen	szTarget 的大小，必须大于或等于 nSourceLen

返回值：

0	转换失败。失败原因可以通过函数fPcm_GetLastErrMsg获得
>0	转换的字节数

功能描述：

将保存在缓冲区中的 8 Bit 无符号 PCM 格式数据转换成 A-law 格式。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_MemAlawToPcm8，fPcm_ALawConvertPcm8

2.26.20 fPcm_MemAlawToUlaw

2.26.21 fPcm_MemUlawtoAlaw

将保存在缓冲区中的 A-Law 格式数据转换成 μ-Law 格式（fPcm_MemAlawToUlaw）。

将保存在缓冲区中的 μ-Law 格式数据转换成 A-Law 格式（fPcm_MemUlawtoAlaw）。

函数原型：

int WINAPI fPcm_MemAlawToUlaw(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)

int WINAPI fPcm_MemUlawtoAlaw(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)

参数说明：

szSource	输入缓冲区的首地址指针，存放 A-Law 或者 μ-Law 格式的语音数据
nSourceLen	szSource 的大小，单位为字节
szTarget	存放转换后的数据的缓冲区指针，由应用程序分配存储空间
nTargetLen	szTarget 的大小，必须大于或等于 nSourceLen

返回值：

0	转换失败。失败原因可以通过函数fPcm_GetLastErrMsg获得
>0	转换的字节数

功能描述：

将保存在缓冲区中的 A-Law 格式数据转换成 μ-Law 格式。

将保存在缓冲区中的 μ-Law 格式数据转换成 A-Law 格式。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.26.22 fPcm_MemPcm16ToAlaw

将保存在缓冲区中的 16 Bit 无符号 PCM 格式数据转换成 A-law 格式。

函数原型：

DWORD fPcm_MemPcm16ToAlaw(char* pSource,DWORD dwSourceSize,char* pTarget, DWORD dwTargetSize)

参数说明：

pSource	输入缓冲区的首地址指针，存放 PCM16 率格式的语音数据
dwSourceSize	pSource 的大小，单位为字节
pTarget	存放转换后的数据的缓冲区指针，由应用程序分配存储空间
dwTargetSize	pTarget 的大小，必须大于或等于 dwSourceSize 的一半

返回值：

0	转换失败。失败原因可以通过函数fPcm_GetLastErrMsg获得
>0	转换的字节数

功能描述：

将保存在缓冲区中的 16 Bit 无符号 PCM 格式数据转换成 A-law 格式。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_MemAlawToPcm16，fPcm_AlawConvertPcm16

2.26.23 fPcm_GC8Convert

将 G.729A 格式的文件转换成其它语音编码格式的文件。

函数原型：

int fPcm_GC8Convert (char*szSoureFile, char* szTargetFile, int nTargetFormat)

参数说明：

szSoureFi	G.729A 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。
ntargetFormat	目标文件的编码格式： =10001：16 Bit PCM =6： A-law

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

将 G.729A 格式的文件转换成其它语音编码格式的文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPCM_AlawConvertGC8

2.26.24 fPcm_Vox6KTo8K

2.26.25 fPcm_Vox8KTo6K

实现 VOX 格式的语音文件在 6K 与 8K 采样率之间进行转换。

函数原型：

int fPcm_Vox6kTo8k(char *sz6kHzFile,char *sz8kHzFile)

int fPcm_Vox8KTo6K (char *sz8kHzFile,char *sz6kHzFile)

参数说明：

sz6kHzFile	采样率为 6kHz、编码格式为 Vox 的语音文件，没有文件头。如果不包含全路径，驱动程序在当前目录下查找文件。
sz8kHzFile	采样率为 8kHz、编码格式为 Vox 的语音文件，没有文件头。如果不包含全路径，驱动程序在当前目录下查找文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

fPcm_Vox6kTo8k 将采样率为 6kHz 的 Vox 文件转换成采样率为 8kHz 的 Vox 文件，fPcm_Vox8KTo6K 将采样率为 8kHz 的 Vox 文件转换成采样率为 6kHz 的 Vox 文件，。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.7 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.26.26 fPcm_ULawConvertGSM

将 μ-Law 格式文件转换为 GSM 格式的文件。

函数原型：

int WINAPI fPcm_ULawConvertGSM(char*szSourceFile, char*szTargetFile)

参数说明：

szSourceFile	μ-Law 格式的源文件，可包括文件的全路径。文件格式为标准的 wav 文件。如果不是，就将它视为无格式文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

1	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

fPcm_ULawConvertGSM 将 μ-Law 格式文件转换为 GSM 格式的文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。
- 如果在驱动没有安装的情况下调用 fPcm_ULawConvertGSM 函数，则不仅需要 ShPcmHandle.dll 文件，还必须使用 macmcvt.dll 文件。

相关信息：

驱动程序版本要求	要求 SynCTI Ver. 4.8.0.0 或更高
Header	ShPcm.h
Library	ShPcmHandle.lib

DLL	ShPcmHandle.dll，macmcvt.dll

相关函数：fPcm_Pcm16ConvertAlaw、fPcm_GC8Convert、fPcm_MemAlawToPcm8、fPcm_MemAlawToPcm16

2.26.27 fPCM_Pcm16ConvertG729A

将 PCM16 格式文件转换为 G.729A 格式文件。

函数原型：

Int WINAPI fPCM Pcm16ConvertG729A(char*szSoureFile, char* szTargetFile)

参数说明：

szSoureFile	用 Synway 板卡录制的 PCM16 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件。
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件。

返回值：

0	调用成功
-1	调用失败。失败原因可以通过函数fPcm_GetLastErrMsg获得。

功能描述：

将 Synway 板卡录制的 PCM16 格式文件转换为 G.729A 格式文件。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcm.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：fPcm_Pcm16ConvertAlaw

2.26.28 fPcm_GetLastErrMsg

获取语音编码格式转换函数调用失败的原因。

函数原型：

int fPcm_GetLastErrMsg(char*szMsg)

参数说明：

szMsg:	用来存放错误信息的指针，空间由应用程序分配

返回值：

0	调用成功
-1	调用失败

功能描述：

获取语音编码格式转换函数调用失败的原因。

注意事项：

- 本函数由ShPcmHandle.dll提供，因此，可以在SsmStartCti之前使用。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高

Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.26.29 fPcm_Close

释放本工具软件所占用的资源。

函数原型：

void fPcm_Close()

参数说明：无

返回值：无

功能描述：

释放本工具软件所占用的资源。

注意事项：

- 使用完本工具中所有转换函数后请务必调用此函数以释放所占用的资源。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：

2.27 有关传真操作的函数（CTI 系列）

2.27.1 设置传真操作的工作参数

2.27.1.1 SsmFaxSetMaxSpeed

设置接收或发送传真时使用的最大速率。

函数原型：

void SsmFaxSetMaxSpeed(int speed)

参数说明：

speed	最大速率，单位为 bps。可选值为 4800 bps、9600bps 和 14400 bps，缺省值为 9600bps。

返回值：无

功能描述：

设置接收或发送传真时使用的最大速率。

注意事项：

本函数设置的参数对所有传真通道都起作用。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxGetSpeed

2.27.1.2 SsmFaxSetID

设置本端的识别代码。

函数原型：

int SsmFaxSetID(int ch,char *szID)

参数说明：

ch	传真通道编号
szID	ASCII 形式的本端识别代码字符串指针。允许长度：少于 20 个字符。

返回值：

-1	调用失败
0	调用成功

功能描述：

设置本端的识别代码。识别代码通常是本端的电话号码，也可以是其它形式的 ASCII 字符。本端识别代码在接收或发送传真时会被传送给对端传真机。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxGetID

2.27.2 发送传真

2.27.2.1 SsmFaxStartSend

2.27.2.2 SsmFaxStartSendEx

启动发送传真文件的任务。SsmFaxStartSend 用于发送整个文件，SsmFaxStartSendEx 用于发送一个文件中的指定页码段。

函数原型：

int SsmFaxStartSend(int ch,char *filename)

int SsmFaxStartSendEx(int ch, char *filename, int nStartPage, int nEndPage)

参数说明：

ch	传真通道编号
filename	tiff文件名，文件的扩展名可以为“.tif”或“.tiff”。有关传真通道支持的文件格式请参见第 1 章中“传真”部分内容。
nStartPage	指定传真文件的起始页。如果 filename 中包含 N 页传真，那么取值范围：1~N
nEndPage	指定传真文件的结束页。如果 filename 中包含 N 页传真，则取值范围： 1~N：指定结束页的编号； ≥N： 发送到第 N 页（含）； -1： 发送到文件的最后一页（即第 N 页） **注意：nEndPage 必须大于或等于 nStartPage**

返回值：

0	函数调用成功
-1	函数调用失败。失败原因可能是由于该通道不处于空闲状态，或打开文件失败。

功能描述：

启动发送传真文件的任务。

发送传真文件的任务启动后，驱动程序开始与对端传真机交换握手信息，握手成功后将传真文件发送到对端传真机。

每当驱动程序完成一页传真的发送，会向应用程序抛出E_CHG_FaxPages事件；当驱动程序完成发送传真的任务后，会向应用程序抛出E_PROC_FaxEnd事件。在发送传真文件的过程中，应用程序可以随时调用函数SsmFaxCheckEnd来检查发送过程是否顺利完成，或者调用SsmFaxStop函数来终止该任务。

注意事项：

- 必须在模拟中继线通道或数字中继线通道的来话或去话呼叫建立后，才能启动发送传真文件的任务；
- 传真通道只能工作在单工方式，即不能同时执行传真接收和发送任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmFaxCheckEnd，SsmFaxStop，SsmFaxSendMultiFile，SsmFaxSendMultiFileEx，SsmFaxAppendSend

2.27.2.3 SsmFaxSendMultiFile

2.27.2.4 SsmFaxSendMultiFileEx

启动发送多个传真文件的任务。SsmFaxSendMultiFileEx 可指定页码范围。

函数原型：

int SsmFaxSendMultiFile(int ch, char * szPath, char * szFile)

int SsmFaxSendMultiFileEx(int ch, FAX_FILE_SCT *pFileNameList, int nNum)

参数说明：

ch	传真通道编号
szPath	传真文件 szFile 的路径。例如，如果将传真文件存放在 C 盘的 FaxFile 目录下，对于 C/C++，szPath 可写为：”C:\\FaxFile\\”。
szFile	传真文件的文件名列表，文件的扩展名可以为“.tif”或“.tiff”，相邻的文件名之间用“;”隔开。例如，要发送的传真文件分别为”A.tif”和”B.tif”，szFile 应该写为：“A.tif;B.tif” **注意：szFile 中的传真文件必须具有相同的格式。**

pFileNameList	传真文件表。结构 FAX_FILE_SCT 的声明为：
	typedef struct tagFAX_FILE_SCT{
	char szFileName[256]; //包含全路径的文件名
	int nStartPage; //起始页，取值范围：1~N
	int nEndPage; //=1~N：指定结束页； =-1：到最后一页
	int nReserve1; //保留
	int nReserve2; //保留
	}FAX_FILE_SCT, *PFAX_FILE_SCT;
	上式中，N 为传真文件中的页面总数。例如，nStartPage=1，nEndPage=1 表示只发第 1 页；nStart=1,nEndPage=2 表示发送第 1 页和第 2 页；如果 nStart=1，nEndPage=-1 表示发送整个文件。
	注意：nEndPage 必须大于或等于 nStartPage

nNum	传真文件的个数

返回值：

0	调用成功
-1	调用失败，函数SsmGetLastErrMsg可获取失败原因

功能描述：

启动发送多个传真文件的任务。

调用本函数后，驱动程序开始与对端传真机交换握手信息，然后按照从左到右的顺序依次将 szFile 中指定的传真文件发送到对端传真机，或从 pFileNameList 表头开始发送文件，一直到最后一个文件发送完毕。

每当驱动程序完成一页传真的发送或接收，会向应用程序抛出E_CHG_FaxPages事件；当驱动程序完成发送传真的任务后，会向应用程序抛出E_PROC_FaxEnd事件。在发送多个传真文件任务的执行过程中，应用程序可以随时调用SsmFaxCheckEnd函数检查任务的完成情况，也可以调用SsmFaxStop来终止该任务。

注意事项：

- 必须在模拟中继线通道或数字中继线通道的来话或去话呼叫建立后，才能启动发送传真文件的任务；
- 传真通道只能工作在单工方式，即不能同时执行传真接收和发送任务；
- 文件路径中不能出现冒号、顿号、句号等标点符号。多文件同时发送时，要求传真文件的格式完全相同；

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxCheckEnd，SsmFaxStop，SsmFaxStartSend，SsmFaxStartSendEx

2.27.2.5 SsmFaxAppendSend

向驱动程序追加一个传真文件。

函数原型：

int SsmFaxAppendSend (int ch,char *filename)

参数说明：

ch	传真通道编号
filename	传真文件名，文件的扩展名可以为“.tif”或“.tiff”。有关传真通道支持的文件格式请参见第 1 章中“传真”部分内容。

返回值：

0	函数调用成功，驱动程序已经把指定的传真文件加入传真发送池。
-1	函数调用失败，失败原因可能是由于该通道处于空闲状态，或打开文件失败。

功能描述：

向驱动程序追加一个传真文件，适用于调用函数SsmFaxSendMultiFile、SsmFaxSendMultiFileEx、SsmFaxStartSend或SsmFaxStartSendEx引发的传真发送任务。

调用本函数后，驱动程序将追加的传真文件加入传真发送池，等当前的传真文件发送完成后，紧接着发送池里的文件。

注意事项：

- 必须在启动传真发送任务后，才能调用本函数；
- 驱动程序在发送传真过程中，如果被用户终止或发送失败，追加的传真文件将被丢弃。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll

相 关 函 数 ： SsmFaxStartSend， SsmFaxStartSendEx， SsmFaxSendMultiFile， SsmFaxSendMultiFileEx，SsmFaxCheckEnd，SsmFaxStop

2.27.3 接收传真

2.27.3.1 SsmFaxStartReceive

启动一次传真接收任务。

函数原型：

int SsmFaxStartReceive(int ch,char *pszFileName)

参数说明：

ch	传真资源通道编号
pszFileName	保存传真数据的 Tiff 格式的文件名，必须以".tif"或".tiff"作为扩展名。如果文件已存在，原有内容将会被覆盖。

返回值：

0	调用成功。
-1	调用失败。失败原因可能是由于该通道不处于空闲状态，或打开文件失败。

功能描述：

启动一次传真接收任务。

调用本函数后，驱动程序将自动开始与对方传真机开始握手过程，握手成功后，开始接收传真内容，并保存到参数 pszFileName 指定的文件中。

每当驱动程序完成一页传真的接收，会向应用程序抛出E_CHG_FaxPages事件；当驱动程序完成全部传真页的接收任务后，会向应用程序抛出E_PROC_FaxEnd事件。在接收传真的过程中，应用程序随时可以调用SsmFaxStop函数终止传真接收任务。

注意事项：

- 传真通道只能工作在单工方式，即不能同时执行传真接收和发送任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxCheckEnd，SsmFaxStop

2.27.4 终止传真任务

2.27.4.1 SsmFaxStop

终止传真接收或发送任务。

函数原型：

int SsmFaxStop(int ch)

参数说明：

ch	传真通道编号

返回值：

-1	调用失败
0	调用成功

功能描述：

终止传真接收或发送任务。

本函数可以终止由函数SsmFaxSendMultiFile，SsmFaxSendMultiFileEx，SsmFaxStartSend，SsmFaxStartSendEx发起的传真发送任务，也可以终止由函数SsmFaxStartReceive发起的传真接收任务。

本函数执行完毕后，驱动程序会向应用程序抛出E_PROC_FaxEnd事件。本函数的执行情况也可以调用函数SsmFaxCheckEnd来获取。

注意事项：

- 本函数是一个异步函数，执行的进展情况可以通过函数SsmFaxCheckEnd获取。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxStartSend，SsmFaxStartSendEx，SsmFaxSendMultiFile，SsmFaxSendMultiFileEx，SsmFaxAppendSend，SsmFaxStartReceive，SsmFaxCheckEnd

2.27.5 获取传真操作的信息

2.27.5.1 SsmFaxGetSpeed

获取传真接收或发送时实际使用的速率。

函数原型：

int SsmFaxGetSpeed(int ch)

参数说明：

ch	传真通道号

返回值：

-1	函数不支持
>0	传真速率，单位为 bps

功能描述：

获取传真接收或发送时实际使用的速率。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxSetMaxSpeed

2.27.5.2 SsmFaxCheckEnd

查询传真接收任务或传真发送任务的进展情况。

函数原型：

int SsmFaxCheckEnd(int ch)

参数说明：

ch	传真资源通道的编号

返回值：

0	任务尚未完成。
1	任务已经完成，传真资源通道已进入“空闲”状态。
2	驱动程序在执行最近一次传真接收或发送任务时发生错误，或者任务被应用程序调用 SsmFaxStop 函数终止，通道已经回到空闲状态。再次调用本函数将得到返回值 1。
3	已经完成传真数据的接收或发送，正在进行断开连接的协商过程。
-1	调用失败

功能描述：

查询传真接收任务或传真发送任务的进展情况。

注意事项：

- 本函数既适用于接收传真任务，也适用于发送传真任务；
- 如果函数返回值为 2，传真任务未完成，但传真数据的接收或发送可能已经完成；
- 如果应用程序采用事件模式进行编程，建议使用 E_PROC_FaxEnd 事件来取代本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： SsmFaxStartSend，SsmFaxStartSendEx，SsmFaxSendMultiFile，SsmFaxStop，SsmFaxSendMultiFileEx，SsmFaxStartReceive，SsmFaxAppendSend

2.27.5.3SsmFaxGetChStateMsg

获取传真资源通道的状态。

函数原型：

int SsmFaxGetChStateMsg(int ch ,char *buf)

参数说明：

ch	传真通道编号
buf	返回状态字符串。buf 的存储空间必须由应用程序分配，并保证大于 100 个字符

返回值：

-1	调用失败
0	调用成功

功能描述：

获取传真资源通道的状态。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.27.5.4 SsmFaxGetPages

获取已经完成的传真接收或发送的页数。

函数原型：

int SsmFaxGetPages(int ch)

参数说明：

ch	传真通道编号

返回值：

-1	调用失败
>0	已完成的传真页数

功能描述：

获取已经完成的传真接收或发送的页数。

注意事项：

- 如果传真接收或发送任务结束后，调用本函数得到的返回值为 0，表明本次传真接收或发送任务完全失败；
- 如果应用程序采用事件模式进行编程，建议使用 E_CHG_FaxPages 事件来取代本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.27.5.5 SsmFaxGetID

获取对端传真机的识别代码。

函数原型：

int SsmFaxGetID(int ch,char *myid)

参数说明：

ch	传真通道编号
myid	存放识别码的缓冲区指针，由应用程序分配存储空间，缓冲区长度不能少于 20 字节

返回值：

-1	调用失败
0	调用成功

功能描述：

获取对端传真机的识别代码。

注意事项：

- 请在握手过程完成才能调用本函数，否则无法得到正确的识别代码。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：SsmFaxSetID

2.27.5.6 SsmFaxGetMode

获取传真通道的工作属性——发送或接受、精细模式或普通模式、ECM 方式或数据流方式。

函数原型：

int SsmFaxGet(int ch, int * pnDir, int *pnResMode, int * pnTransMode)

参数说明：

Ch	传真通道编号
pnDir	存放发送或接收指示值 =0：表示传真通道进行接收工作 =1：表示传真通道进行发送工作
pnResMode	存放传真数据模式的指示值 =0：表示普通模式 =1：表示精细模式
pnTransMode	存放传真通道传输模式的指示值 =0：表示数据流模式 =1：表示 ECM 模式

返回值：

-1	通道不支持此操作
0	传真通道空闲，本函数取不到所需参数
1	函数调用成功

功能描述：

获取传真通道的工作属性——发送或接受、精细模式或普通模式、ECM 方式或数据流方式。

注意事项：

- 只有在握手过程完成之后才能调用本函数，否则无法得到正确的识别代码。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.27.5.7 SsmFaxGetAllBytes

发送传真时，获取当前传真发送页所包含的总字节数。

函数原型：

int SsmFaxGetAllBytes(int ch)

参数说明：

ch	传真通道号

返回值：

-1	调用失败
≥0	当前传真页的总字节数

功能描述：

发送传真时，获取当前传真发送页所包含的总字节数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.27.5.8 SsmFaxGetSendBytes

发送传真时，获取当前发送页中已经完成发送的字节数。

函数原型：

int SsmFaxGetSendBytes(int ch)

参数说明：

ch	传真通道编号

返回值：

-1	调用失败
≥0	当前页已发送的字节数

功能描述：

发送传真时，获取当前发送页中已经完成发送的字节数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.27.5.9 SsmFaxGetRcvBytes

接收传真时，获取当前接收页中已经完成接收的字节数。

函数原型：

int SsmFaxGetRcvBytes(int ch)

参数说明：

ch	传真通道编号

返回值：

-1	调用失败
≥0	当前页已接收的字节数

功能描述：

接收传真时，获取当前接收页中已经完成接收的字节数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.27.6 .tif 文件的相关操作

2.27.6.1 fBmp_ValidateFaxFile

判断.tif 文件是否为符合三汇板卡的传真格式。

函数原型：

int fBmp_ValidateFaxFile(char *szFile)

参数说明：

szFile	存放传真文件的字符串

返回值：

0	调用成功，传真发送支持此.tif 文件
-1	传真发送不支持此.tif文件，具体原因可从fBmp_GetErrMsg获得
-2	打开文件失败，具体原因可从fBmp_GetErrMsg获得

功能描述：

判断.tif 文件是否为符合三汇板卡的传真格式。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：

2.27.6.2 fBmp_SetHeaderFormat

设置传真.tif 文件页眉属性

函数原型：

int fBmp_SetHeaderFormat(int nRow, int nFromX, int nFromY, char szFrom, int nSubX, int nSubY, char*szSubject, int nToX, int nToY, char*szTo, int nTimeX, int nTimeY, char*szTime)

参数说明：

nRow	表示页眉的行数，1~3 之间范围，默认为 1
nFromX	表示 From 字段的起始位置， 默认为 5
nFromY	表示 From 所在的行数，1~3 之间，1 表示第一行，默认为 1
szFrom	表示 From 提示字段，默认为"From:"，最大可为 20byte
nSubX	表示 subject 字段的起始位置，默认为 270
nSubY	表示 Subject 所在的行数，1~3 之间，1 表示第一行，默认为 1
szSubject	表示 Subject 提示字段，默认为"Sub:"，最大可为 20byte
nToX	表示 To 字段的起始位置，默认为 440
nToY	表示 To 所在的行数，1~3 之间，1 表示第一行，默认为 1
szTo	表示 To 提示字段，默认为"To:"，最大可为 20byte
nTimeX	表示 Time 字段的起始位置，默认为 660
nTimeY	表示 Time 所在的行数，1~3 之间，1 表示第一行，默认为 1
szTime	表示 Time 提示字段，默认为""，最大可为 20byte

返回值：

-1	调用失败 ，失败原因可从fBmp_GetErrMsg获得
1	操作成功

功能描述：

设置 fBmp_AddTxtToTif 所生成页眉的属性。

注意事项：

- 如果要将 Subjuect 放在第二行，那么将 nRow =2, nSubY=2 即可；
- 横向位置的范围是 0~864。可根据需要调整各段数据的位置，注意不要越出边界。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高

Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：fBmp_AddTxtToTif，fBmp_AddTxtToTif_Big，fBmp_UniteTif

2.27.6.3 fBmp_AddTxtToTif

增加页眉到.tif 文件上，只支持国际码编码。

函数原型：

int fBmp_AddTxtToTif(char szTifName, char* szFaxFrom, char*szFaxTo, char *szFaxSubbject, char * szDataTime , char *szTargetFile, DWORD dwReserve)

参数说明：

szTifName	需要加页眉的.tif 文件
szFaxFrom	传真发送方的信息，最大长度为 20byte，1 个汉字占 2byte.
szFaxTo	传真接收方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxSubject	传真主题，最大占 10byte，1 个汉字占 2byte
szDataTime	时间信息，最大占 20byte
szTargetFile	要生成带页眉的 tif 文件，如果与 szTifName 重名则覆盖原文件
dwReserve	保留参数，目前无意义，以后扩展使用，使用时设置为 0

返回值：

-1	调用失败 ，失败原因可从fBmp_GetErrMsg获得
1	操作成功

功能描述：

增加页眉到.tif 文件上。在 szTifName 文件顶部增加 16 行、每行 1728 像素的页眉数据，把结果输出到 szTargetFile 文件。当设置 dwReserve 的第 2 比特位为 1 时，dwReserve 的高 16bit 表示需要加页眉的页。

注意事项：

- 如果 szTifName 与 szTargetFile 相同，则覆盖源文件；
- 顶部数据排列从左到右为 szFaxFrom、szSubject、szFaxTo、szDataTime。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：fBmp_SetHeaderFormat，fBmp_AddTxtToTif_Big，fBmp_CutTifHeader

2.27.6.4 fBmp_AddTxtToTif_Big

增加页眉到.tif 文件上，而且在国际码的基础上增加支持大五码繁体字体编码。

函数原型：

int fBmp_AddTxtToTif_Big(char*szTifName, char*szFaxFrom, char*szFaxTo, char*szFaxSubject, Char *szDataTime, char *szTargetFile, DWORD dwReserve)

参数说明：

szTifName	需要加页眉的 tif 文件
szFaxFrom	传真发送方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxTo	传真接收方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxSubject	传真主题，最大占 10byte，1 个汉字占 2byte
szDataTime	时间信息，最大占 20byte

szTargetFile	要生成带页眉的 tif 文件，如果与 szTifName 重名则覆盖原文件
dwReserve	保留参数，目前无意义，以后扩展使用，使用时设置为 0

返回值：

-1	调用失败 ，失败原因可从fBmp_GetErrMsg获得
1	操作成功

功能描述：

增加页眉到.tif 文件上，而且支持大五码繁体字体。在 szTifName 文件顶部增加 16 行，每行 1728 像素的页眉数据，把结果输出到 szTargetFile 文件。当设置 dwReserve 的第 2 比特位为 1 时，dwReserve 的高 16bit 表示需要加页眉的页。

注意事项：

- 如果 szTifName 与 szTargetFile 相同，则覆盖源文件；
- 顶部数据排列从左到右为 szFaxFrom、szSubject、szFaxTo、szDataTime；
- 函数支持操作系统支持的字体。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：fBmp_SetHeaderFormat，fBmp_AddTxtToTif，fBmp_CutTifHeader

2.27.6.5 fBmp_UniteTif

实现两个.tif 文件的叠加。

函数原型：

int fBmp_UniteTif(char*szHeadTif, char * szSourceTif, char *szTargetTif, DWORD dwReserve)

参数说明：

szHeadTif	用来加头的相对较小的.tif 文件，只能是单页文件
szSourceTif	用来被加头的.tif 文件，可以是多页文件
szTargetTif	生成的.tif 文件
dwReserve	保留参数

返回值：

-1	调用失败 ，失败原因可从fBmp_GetErrMsg获得
1	操作成功

功能描述：

实现两个.tif 文件的叠加，将 szHeadTif 文件加到 szSourecTif 文件上，生成 szTargetTif 文件。

注意事项：

- 如果 szTifName 与 szTargetFile 相同，则覆盖源文件；
- 如果 dwReserve 的第 2 Bit 为 1，表示 dwReserve 的高 16 bit 为需要叠加的页面。如果不指定该文件，则 SzSourceTif 的所有页的头上都要加 szHeadTif。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：fBmp_SetHeaderFormat

2.27.6.6 fBmp_CutTifHeader

实现.tif 文件的页眉截取功能。

函数原型：

int fBmp_CutTifHeader(char *Source, char*szTarget, int nHeight, int nPgeNo, DWORD dwReserve)

参数说明：

szSource	需要将页眉截掉的.tif 源文件
szTarget	生成的目标文件，如果文件已经存在则覆盖原有文件
nHeight	需要截掉的高度
nPageNo	.tif 文件中需要截掉页眉的指定页
dwReserve	保留参数

返回值：

-1	调用失败 ，失败原因可从fBmp_GetErrMsg获得
1	操作成功

功能描述：

实现.tif 文件的页眉截取功能，从.tif 文件顶部截取设定的相应部分。

注意事项：

- 如果 szSourceTif 与 szTargetFile 相同，则覆盖源文件 szSourceTif；
- nHeight 一般在 16~40 之间，单行的页眉为 20 左右；
- nPageNO 为指定页，如果设置为-1，则对所有页都截掉页眉。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：fBmp_AddTxtToTif_Big，fBmp_AddTxtToTif

2.27.6.7 fBmp_GetFileAllPage

输出传真文件的页数。

函数原型：

int fBmp_GetFileAllPage(char *filename)

参数说明：

filename	存放传真文件的字符串

返回值：

-1	操作失败，失败原因可能是文件打开失败，或文件格式不对
>=0	操作成功，输出文件的页数

功能描述：

输出传真文件 filename 的页数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.4.0.6 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：

3 SynCti 驱动程序的配置

3.1 系统配置文件 ShConfig.ini

使用 Synway 板卡时，必须对板卡及驱动程序进行正确和必要的配置后，应用系统才能正常工作。

SynCTI 驱动程序的配置信息采用文本方式保存在 INI 文件中。在正确完成 SynCTI 驱动程序的安装后，安装程序会在安装目录下生成一个文件名称为 ShConfig.ini 的配置文件范本。开发人员可以通过文本编辑软件直接编写或修改配置文件，也可以通过 Synway 公司提供的系统配置工具软件来修改配置。

配置文件中，配置信息以节(Section)为单位，每个节可以包含一个或多个配置项(Item)。节的名称出现在一对方括号中，后续的每一个文本行就是一个配置项。节的作用范围为从本节的后面一行开始，直到下一个节的上面一行止。每一个配置项都具有“x=y”的形式，等号左边是配置项的名称，右边是配置内容。

3.1.1 必须配置项

3.1.1.1 设置板卡总数

配置项	**TotalBoards**
节	**[SystemConfig]**
书写格式	TotalBoards=N
取值范围	N 等于系统中安装的 Synway 板卡数量
功能描述	设置系统中安装的 Synway 板卡总数

配置项	**WhoSupplySysClock**
节	**[SystemConfig]**
书写格式	WhoSupplySysClock=n
取值范围	n=-1： 将全部 Synway 板卡均设置为从卡； n=N： 将全部 Synway 板卡均设置为主卡； 0≤n≤N-1：将逻辑编号为 n 的板卡设置为主卡，其余板卡均为从卡。 注：N为配置项TotalBoards的设定值。
功能描述	设置应用系统中的Synway板卡的主卡。更多信息请参见第 1 章中“配置应用系统的时钟”部分内容。

3.1.1.2 设置通道编号

配置项	**TotalAppCh**
节	**[AppChToBoardChTable]**
书写格式	TotalAppCh=M
取值范围	M 为系统中全部板卡上的通道总数
功能描述	设置系统中的通道总数

配置项	**AppCh**
节	**[AppChToBoardChTable]**

书写格式	格式 1：AppCh[i_s]=k,j_s…j_e 格式 2：AppCh[i]=k,j 上式中， i： 应用程序使用的通道号，通常从 0 开始编号 i_s：应用程序使用的起始通道号 k：需要建立映射的板卡 BoardID j： 通道在板卡内部的编号 j_s：通道在板卡内部的起始编号 j_e：通道在板卡内部的结束编号
取值范围	
功能描述	建立应用程序的通道编号与板卡通道编号之间的映射。格式 1 用于批量建立映射，格式 2 用于逐个建立。

3.1.1.3 设置板卡信息

配置项	**BoardModel**
节	**[BoardId=x]**
书写格式	BoardModel=s
取值范围	s是表示板卡型号的字符串。可用的型号请参见第 1 章中Synway板卡的分类部分。
功能描述	设置板卡型号
使用范例	BoardModel=SHT-16B-CT/PCI

配置项	**BoardSerialNumber**
节	**[BoardId=x]**
书写格式	BoardSerialNumber=N
取值范围	N 是板卡的序列号
功能描述	设置板卡的序列号，具体取值请参见相关的硬件说明书。
使用范例	BoardSerialNumber=11

3.1.1.4 SHD/DTP 系列板卡

3.1.1.4.1 设置板卡复位功能

配置项	**ResetBoardOnClose**
节	**[BoardId=x]**
书写格式	ResetBoardOnClose =n
取值范围	n=0：不复位； n=1：复位（缺省值）。
功能描述	设置在关闭板卡时是否将板卡复位。

3.1.1.4.2 设置数字中继线的工作参数

配置项	**PcmNumber** **PcmSSx** **PcmClockMode** **PcmLinkType** **CRC-4**
节	**[BoardId=x]**

书写格式	PcmNumber=M PcmSSx[i]=n PcmClockMode[i]=m PcmLinkType[i]=k CRC-4[i]=c　　//注：要求 SynCTI Ver. 4.7.1.5 及以上版本
取值范围	M：板卡上数字中继线的总数，与板卡型号有关，详细信息请参见第 1 章中“SHD系列板卡简介”部分内容； i：　数字中继线在板卡上的物理编号，更多信息请参见相关手册的说明。取值范围：0≤i≤M； n：选择信令方式。取值范围： n=0：ISDN 协议（用户侧）； n=1：No.1 信令（SS1）； n=2：ISDN 协议（网络侧） n=7：SS7 信令（SS7）； m：设置本数字中继线的时钟模式，具体为： m=0：主时钟、线路同步方式（板卡必须为主卡）； m=1：主时钟、自振荡方式（板卡必须为主卡）； m=2：从时钟 k：选择线路的传输类型。取值范围： k=0：120 欧姆双绞线； k=1：75 欧姆同轴电缆 c：CRC-4 校验功能的控制开关： c=0：关闭 c=1：启用（缺省值）
功能描述	PcmNumber 设置板卡上数字中继线的总数； PcmSSx 设置数字中继线的信令类型； PcmClockMode设置时钟的工作模式，更多信息请参见第 1 章中“设置应用系统的时钟”部分内容； PcmLinkType 设置线路的传输类型； CRC-4 设置板卡上数字中继线的 CRC-4 校验功能。
注意事项	● 如果 PcmNumber 的设定值大于 0，配置项 PcmSSx、PcmClockMode、PcmLinkType 应从 i=0 开始，重复 M 次，即需要对每一条物理数字中继线进行配置。 ● 这些配置项只适用于 SHD/DTP 系列板卡。
使用范例	假设系统中安装了 2 片 SHD-60A-CT/PCI/SS7 板卡，0＃板卡使用 SS7 信令，采用 120 欧姆的双绞线接入，它的 0＃数字中继线采用线路同步方式为应用程序提供主时钟；1＃板卡使用 ISDN 协议（用户侧），采用 75 欧姆同轴电缆接入，则配置项的设置为： **[BoardId=x]** **……** PcmNumber=4 PcmSSx[0]=7　　//SS7 信令方式 PcmSSx[1]=7 PcmSSx[2]=0　　//ISDN 用户侧 PcmSSx[3]=0 PcmClockMode[0]=0　//主时钟，线路同步方式 PcmClockMode[1]=2　//必须设置为从时钟 PcmClockMode[2]=2 PcmClockMode[3]=2 PcmLinkType[0]=0　　//120 欧姆双绞线 PcmLinkType[1]=0 PcmLinkType[2]=1　　//75 欧姆同轴电缆 PcmLinkType[3]=1

3.1.1.4.3 设置 PCM 的逻辑编号

配置项	**TotalPcm** **Pcm**
节	**[PcmInfo]**
书写格式	TotalPcm=M Pcm[m]=k,j
取值范围	M：应用系统中数字中继线的总数，必须大于或等于系统中的物理 PCM 线路的总数 m：应用程序中的数字中继线的逻辑编号，必须从 0 开始编号，取值范围： 0≤m<M (M 为配置项 TotalPcm 的设定值)； k：板卡的逻辑编号。取值范围： ✧ 0≤k<N ：N为配置项TotalBoards的设定值。将应用程序的数字中继线的逻辑编号与实际的物理数字中继线进行绑定。 ✧ k=-1： 将应用程序的数字中继线的逻辑编号与虚电路进行绑定，只适用于SS7 信令的TUP协议和ISUP协议，要求驱动程序的版本为 4.7.1.0 或更高。有关虚电路的更多信息请参见第 1 章中“使用SS7 服务器的虚电路编程接口”部分内容。 j： 如果 k≠-1，j 表示数字中继线的物理编号，必须是物理上存在的 PCM；如果 k=-1，j 所对应的数字中继线可以是虚拟的
功能描述	TotalPcm 设置应用程序中数字中继线的总数； Pcm 建立数字中继线的逻辑编号与物理编号之间的映射关系
注意事项	本配置项只适用于 SHD、DTP 系列板卡

使用范例	假设系统中安装了 2 片 SHD-60A-CT/PCI/ISDN 板卡，则需要按照下列方法建立数字中继线的逻辑编号与物理编号之间的映射：
	[PcmInfo]
	TotalPcm=4
	Pcm[0]=0,0 //将 0#板卡上的 0#数字中继线设置为应用程序的 0#数字中继线
	Pcm[1]=0,1 //将 0#板卡上的 1#数字中继线设置为应用程序的 1#数字中继线
	Pcm[2]=1,0 //将 1#板卡上的 0#数字中继线设置为应用程序的 2#数字中继线
	Pcm[3]=1,1 //将 1#板卡上的 1#数字中继线设置为应用程序的 3#数字中继线

3.1.1.4.4 设置 SpyPCM 编号（DTP 系列）

配置项	**TotalSpyPcm** **SpyPcm**
节	**[SpyPcm]**
书写格式	TotalSpyPcm=M SpyPcm[m]=Pcm[j],Pcm[k]
取值范围	M： 应用系统中需要监控的数字中继线的总数 m： SpyPcm 的编号，从 0 开始编号，0≤m≤M; j,k： 数字中继线的逻辑编号，由**[PcmInfo]**节中的配置项Pcm中设置，更多信息请参见第 1 章中“数字中继线”部分内容。建议用板卡上的 2 条编号连续的数字中继线接入端口来监控一条 SpyPCM，因此，在大多数情况下，k=j+1。注意： j和k必须位于同一片板卡上。
功能描述	TotalSpyPcm 设置应用系统中需要监控的数字中继线的总数； SpyPcm 设置用板卡上的哪 2 条数字中继线接入电路来处理被监控的数字中继线，即将第 m 条 SpyPCM 绑定到第 j 条和第 k 条数字中继线处理电路。 更多信息请参见第 1 章中“DTP系列板卡”部分内容。
注意事项	本配置项只适用于 DTP 系列板卡

使用范例	假设系统中安装了 1 片 SHD-60A-CT/PCI/FJ 板卡，用来采集 2 条使用 75 欧姆同轴电缆的 E1 中继线上的 60 个语音信道（30×2）和 2 个 ISDN PRI 信令信道，则需要按照下列方法进行配置：
	
	[BoardId=x]
	……
	PcmNumber=4
	
	PcmSSx[0]=0 //ISDN 用户侧
	PcmSSx[1]=0
	PcmSSx[2]=0
	PcmSSx[3]=0
	
	PcmClockMode[0]=0 //主时钟，线路同步方式
	PcmClockMode[1]=2 //必须设置为从时钟
	PcmClockMode[2]=2
	PcmClockMode[3]=2
	
	PcmLinkType[0]=1 //75 欧姆同轴电缆
	PcmLinkType[1]=1
	PcmLinkType[2]=1
	PcmLinkType[3]=1
	
	[PcmInfo]
	TotalPcm=4
	
	Pcm[0]=0,0 //将 0#板卡上的 0#数字中继线设置为应用程序的 0#数字中继线
	Pcm[1]=0,1 //将 0#板卡上的 1#数字中继线设置为应用程序的 1#数字中继线
	Pcm[2]=0,2 //将 0#板卡上的 2#数字中继线设置为应用程序的 2#数字中继线
	Pcm[3]=0,3 //将 0#板卡上的 3#数字中继线设置为应用程序的 3#数字中继线
	
	[SpyPcm]
	TotalSpyPcm=2
	
	SpyPcm[0]=Pcm[0],Pcm[1] //用 0#、1#PCM 来采集 0#SpyPCM 上的信息
	SpyPcm[1]= Pcm[2],Pcm[3] //用 2#、3#PCM 来采集 1#SpyPCM 上的信息
	
	[AppSpyCICTable]
	TotalAppSpyCIC=60
	
	AppSpyCIC[0]=SpyPcm[0],0..29 //设置 SpyCIC 逻辑编号为 0~29 的物理位置
	AppSpyCIC[30]=SpyPcm[1],0..29 //设置 SpyCIC 逻辑编号为 30~59 的物理位置

3.1.1.4.5 设置 SpyCIC 编号（DTP 系列）

配置项	**TotalAppSpyCIC** **AppSpyCIC**
节	**[AppSpyCICTable]**
书写格式	TotalAppSpyCIC=N 格式 1：AppSpyCIC[n]=SpyPcm[m],k 格式 2：AppSpyCIC[n]=SpyPcm[m],k_s..k_e
取值范围	N：SpyCIC 的总数 n：SpyCIC 的逻辑编号，从 0 开始编号，0≤n<N m：SpyPCM的逻辑编号，详细内容请参见配置项TotalSpyPcm或SpyPcm的说明 k：SpyPcm 上的电路编号，0≤k≤29。对于 SS7 信令，k 的值就是 TUP 或 ISUP 消息中的 CIC 字段；对于 ISDN PRI 信令，k 的值与 PCM 中的实习关系为： k_s：SpyPcm 上的起始电路编号，0≤k_s≤29 k_e：SpyPcm 上的结束电路编号，0≤k_e≤29

功能描述	TotalAppSpyCIC 设置被监控电路的总数； AppSpyCIC建立应用程序中SpyCIC的逻辑编号与SpyPCM上某个具体电路之间的映射。更多信息请参见第 1 章中“DTP系列板卡”部分内容。
注意事项	● 格式 1 用于逐个设置，格式 2 用于批量设置相当于从 k_s 开始，用格式 1 连续设置 k_e-k_s+1 次。 ● 本配置项只适用于 DTP 系列板卡
使用范例	请参见配置项TotalSpyPcm中的使用范例

3.1.1.4.6 SS7 信令的必须配置项（DTP 系列）

配置项	**TotalSpyLinkSet** **SpyLinkSet** **SpyCICPcm** **SpyIsupCICPcm**
节	**[SS7Spy]**
书写格式	TotalSpyLinkSet=N 格式 1：SpyLinkSet[n]=SpyPcm[x] 格式 2：SpyLinkSet[n]=SpyPcm[x]+SpyPcm[y] SpyLinkSet[n].SpyCICPcm[k]=SpyPcm[m] SpyLinkSet[n].SpyIsupCICPcm[k]=SpyPcm[m] **注：格式 1 适用于链路组中只有 1 条信令链路的情况，格式 2 适用于链路组中有 2 条信令链路的情况**
取值范围	N： 信令链路组的总数，1≤N≤48，必须对从 n=0 开始，将 SpyLinkSet 连续配置 N 次。 n： 链路组的逻辑编号，0≤n<N x,y, m：SpyPcm 的逻辑编号，0≤x<M，0≤y<M，0≤m<M k： 电信局为 SpyPcm 分配的数字中继线编号，即 TUP 或 ISUP 消息中 CIC 字段中 PCM 参数的值，具体值可以向电信局获取。 **注**：M为配置项TotalSpyPcm的设定值
功能描述	✧ TotalSpyLinkSet 设置 SS7 信令链路组的总数； ✧ SpyLinkSet 设置指定第 n 个链路组中信令链路的物理位置； ✧ SpyCICPcm、SpyIsupCICPcm 建立第 n 个链路组中 TUP 或 ISUP 消息的 CIC 字段的数字中继线编号与 SpyPCM 的逻辑编号之间的映射关系，SpyCICPcm 只适用于 TUP 协议，SpyIsupCICPcm 只适用于 ISUP 协议。
注意事项	● SpyLinkSet[n].SpyCICPcm[k]只适用于 TUP 协议； ● SpyLinkSet[n].SpyIsupCICPcm[k]只适用于 ISUP 协议； ● 本配置项只适用于 DTP 系列板卡的 SS7 信令； ● 本配置项要求 SynCTI 驱动程序的版本为 4.5.8.0 或更高。
使用范例	请参见第 1 章中“DTP系列板卡的应用及配置实例”部分内容。

3.1.1.4.7 SS7 信令的必须配置项（SHD 系列）

3.1.1.4.7.1 设置使用 ISUP 协议的数字中继线

配置项	**TotalIsupPcm** **IsupPcm**
节	**[SS7]**
书写格式	TotalIsupPcm=M IsupPcm[m]=LocalPcm[k]

取值范围	M：使用ISUP协议的数字中继线的总数，0≤M<N，N为配置项TotalPcm的设定值。如果M不为 0，必须设置配置项IsupPcm。 m：使用 ISUP 协议的数字中继线的逻辑编号，从 0 开始编号。取值范围：0≤x<M k：本应用系统中的数字中继线的逻辑编号，取值范围：0≤k<N。数字中继线的逻辑编号由配置项TotalPcm和Pcm设置。
功能描述	TotalIsupPcm 设置应用系统中使用 ISUP 协议的数字中继线的总数； IsupPcm 设置本机中使用 ISUP 协议的数字中继线
注意事项	● 如果系统中没有使用 ISUP 协议，则不需要配置本配置项； ● 如果TotalIsupPcm设置为 0，则不需要设置配置项IsupPcm，并且配置项AutoHandleIsup将被忽略； ● 本配置项只适用于 SHD 系列板卡

3.1.1.4.8 ISDN 信令必须配置项（SHD 系列）

3.1.1.4.8.1 设置运行模式

配置项	**UseISDNMode**
节	**[ISDN]**
书写格式	UseISDNMode=m
取值范围	m=0：用户侧方式（缺省值）； m=1：网络侧； m=2：用户侧＋网络侧。**注意：在此模式下，同一片板卡上的全部数字中继线必须具有相同的模式，即或者都是用户侧，或者都是网络侧，但不同板卡可以具有不同的模式。**
功能描述	设置使用ISDN协议的模式。更多信息请参见第 1 章中“网络侧模式与用户侧模式”部分内容。

3.1.1.4.8.2 设置使用 ISDN 信令(用户侧)的数字中继线

配置项	**TotalUserLinker** **UserPcmLink** **UserTEIValue**
节	**[ISDN]**
书写格式	TotalUserLinker=N UserPcmLink[n]=LocalPCM[k] UserTEIValue[n]=m
取值范围	N：系统中用户侧数字中继线的总数； n：用户侧数字中继线的逻辑编号，从 0 开始编号； k：数字中继线在本端的逻辑编号，详细内容请参见配置项TotalPcm和Pcm的说明； m：TEI 值。可取的值为： 0≤m≤63：固定 TEI； m>63： 动态 TEI。 m 的缺省值为 0。 **注意：m的取值必须与网络侧保持一致，否则会导致本端与网络侧之间无法建立链路。假设网络侧的TEI值为x，如果 0≤x≤63，则m必须等于x；如果x>63，则m可以等于x，也可以不必等于x。有关TEI值的更多信息，请参见第 1 章中“TEI值”部分内容。**
功能描述	TotalUserLinker 设置本端使用 ISDN 信令、并且工作于用户侧的数字中继线的总数； UserPcmLink 建立每条用户侧数字中继线的编号与其逻辑编号之间的映射关系； UserTEIValue 设置用户侧数字中继线的 TEI 值。
注意事项	● 如果 N>0，UserPcmLink 和 UserTEIValue 都必须设置 N 次； ● 本配置项只适用于用户侧模式

3.1.1.4.8.3 设置使用 ISDN 信令(网络侧)的数字中继线

配置项	**TotalNetLinker** **NetPcmLink** **NetTEIValue**
节	**[ISDN]**
书写格式	TotalNetLinker=N NetPcmLink[n]=LocalPCM[k] NetTEIValue[n]=m
取值范围	N： 系统中网络侧数字中继线的总数； n： 网络侧数字中继线的逻辑编号，从 0 开始编号； k： 数字中继线在本端的逻辑编号，详细内容请参见配置项TotalPcm和Pcm的说明； m：TEI 值。可取的值为： 0≤m≤63：固定 TEI； m>63： 动态 TEI。 m 的缺省值为 0。
功能描述	TotalNetLinker 设置本端工作于网络侧的数字中继线的总数； UserPcmLink 建立每条网络侧数字中继线与其逻辑编号之间的映射关系； UserTEIValue 设置每条网络侧数字中继线的 TEI 值。
注意事项	● 如果 N>0，NetPcmLink 和 NetTEIValue 都必须设置 N 次； ● 本配置项只适用于用户侧模式。

3.1.1.4.9 SS1 信令的必须配置项（SHD 系列）

配置项	**ProtocolType**
节	**[SS1Config]**
书写格式	ProtocolType=m
取值范围	m=0：使用 1 号信令（缺省值）； m=1：使用 LineSide 协议。
功能描述	设置 SS1 通道的信令方式 。

3.1.1.5 SHN 系列板卡

3.1.1.5.1 设置 IP 通道的工作参数

配置项	**ProtocolType** **RecvDtmfType**
节	**[BoardId=x]**
书写格式	ProtocolType=m RecvDtmfType=k
取值范围	m：设置本 IP 卡使用的协议类型，具体为： m=0：使用 H.323 类型； m=1：使用 SIP 类型。 k：选择接收 DTMF 的方式。取值范围： k=0：接收采用信令、带内、RFC2833 事件方式发送的 DTMF； k=1：仅接收采用信令和 RFC2833 事件方式发送的 DTMF； k=2：仅接收采用带内方式发送的 DTMF。
功能描述	ProtocolType 设置本 IP 卡使用的通讯协议； RecvDtmfType 设置本 IP 卡接收 DTMF 的方式。

3.1.1.5.2 SIP 信令的必须配置项（SHN 系列）

配置项	**LocalIp** **SendDtmfType**
节	**[SIP]**
书写格式	LocalIp=m SendDtmfType=j
取值范围	m：设置 SIP 信令承载的网络地址。m 为字符串，取值形式为 Ipv4 地址，长度小于 256 个字符。特殊情况： m=0.0.0.0：随机选择本地主机 IP 地址之一作为 SIP 信令承载的网络地址。 j：设置发送 DTMF 的方式，取值范围： j=0：采用 RFC2833 事件方式发送 DTMF； j=1：采用信令方式发送 DTMF； j=2：采用带内方式发送 DTMF。
功能描述	设置 SIP 信令承载的网络地址、发送 DTMF 方式

3.1.1.5.3 H.323 信令的必须配置项（SHN 系列）

配置项	**LocalIp** **GlobalName** **SendDtmfType**
节	**[H.323]**
书写格式	LocalIp=m GlobalName=k SendDtmfType=j
取值范围	m：设置 H.323 信令承载的网络地址。m 为字符串，取值形式可以是 Ipv4 地址或主机名，长度小于 256 个字符。特殊情况： m=0.0.0.0：使用本地主机所有 IP 地址作为 H.323 信令承载的网络地址(多穴)。 k：设置全局呼叫名称，被当作主叫号码发送给被叫，k 是字符串，长度小于 256 个字符。 j：设置发送 DTMF 的方式，取值为： j=0：采用 RFC2833 事件方式发送 DTMF； j=1：采用信令方式发送 DTMF； j=2：采用带内方式发送 DTMF。
功能描述	设置 H.323 侦听地址、全局呼叫名称、发送 DTMF 方式

3.1.1.6 DST 系列板卡（REC 系列 only）

配置项	**PBXType**
节	**[BoardId=x]**
书写格式	PBXType=s
取值范围	s：表示交换机型号的字符串，可选的型号请参见第 1 章中“DST系列板卡支持的交换机型号”部分内容。
功能描述	设置 DST 系列板卡所连接的交换机型号。
使用范例	PBXType=ALCATEL4300

配置项	**PhoneType**
节	**[BoardId=x]**
书写格式	PhoneType=n_1,n_2,n_3,n_4,n_5,n_6,n_7,n_8,n_9,n_{10},n_{11},n_{12},n_{13},n_{14},n_{15},n_{16}
取值范围	n_1~n_{16}：表示数字电话机型号的数字，可选值请参见第 1 章中“DST系列板卡支持的交换机型号”部分内容。

功能描述	设置 DST 板卡所连接的数字电话机型号的设置值。

3.1.2 常用配置项

3.1.2.1 设置 SHD/DTP 系列板卡的协议

配置项	**CardType**
节	**[BoardId=x]**
书写格式	CardType=n
取值范围	n=0：E1 接口 n=1：T1 接口 n=2：J1 接口 缺省值为 0。
功能描述	通过该配置项和 PcmSSx 配置项，使系统可以在 T1/J1 ISDN 环境下使用。同时，您需要根据线路语音编码格式，修改以下配置项： 1.[BoardId=x]节语音DSP接收/发送语音编码配置项DefaultVoiceFormat； 2.若为ISDN线路，则修改[ISDN] 节语音编码格式配置项UserVoiceFormat或NetVoiceFormat。

3.1.2.2 设置驱动程序抛出的事件

配置项	**DefaultEventOutput**
节	**[SystemConfig]**
书写格式	DefaultEventOutput=m
取值范围	m=0：不输出任何事件 m=1：只输出常用事件。有关常用事件的更多信息请参见第 1 章中“驱动程序抛出的事件”部分内容。 m=2：输出全部事件 缺省值为 1。
功能描述	设置驱动程序的事件过滤器。更多信息请参见第 1 章中“事件的过滤”部分内容。

3.1.2.3 设置事件的数据结构

配置项	**EventInterfaceType**
节	**[SystemConfig]**
书写格式	EventInterfaceType=n
取值范围	n=0：缺省值。使用 MESSAGE_INFO，此时函数 SsmWaitForEvent、SsmGetEvent、SsmPutUserEvent有效； n=1：使用SSM_EVENT，此时函数SsmWaitForEventA、SsmGetEventA、SsmPutUserEventA有效。
功能描述	当驱动程序将事件输出到事件队列时，本配置项决定事件使用的数据结构。
注意事项	如果使用 DST 系列板卡，并且需要使用事件 E_RCV_DSTDChannel，本配置项必须配置为 1。

3.1.2.4 设置驱动程序内部事件队列的深度

配置项	**MaxEventPerChannel**
节	**[SystemConfig]**
书写格式	MaxEventPerChannel=n
取值范围	n>0，单位为事件的个数，缺省值为 100
功能描述	设置一个通道的事件队列的深度。当应用程序以wEvent=0xffff作为参数调用SsmSetEvent函数来启用事件等待或事件回调模式时，驱动程序会使用本配置项的设定值为事件队列分配内存。

配置项	**MaxUserEventSize**
节	**[SystemConfig]**
书写格式	MaxUserEventSize=n
取值范围	n≥0，缺省值为 0（表示禁止应用程序向驱动程序添加自定义事件）

功能描述	在驱动程序的一次硬件中断时间间隔（8ms）内，应用程序可以向驱动程序内部的事件输出队列中添加的自定义事件的最大数量

3.1.2.5 设置状态机的通用参数

配置项	**MaxWaitAutoDialAnswerTime**
节	**[SystemConfig]**
书写格式	MaxWaitAutoDialAnswerTime=t
取值范围	t：最大等待时间，单位为秒，缺省值因通道类型而异，具体为： ✧ 模拟中继线通道：25 秒； ✧ SS1 通道：90 秒； ✧ ISUP 通道：180 秒； ✧ TUP 通道：60 秒。
功能描述	在去话呼叫过程中，当本端完成全部被叫号码的发送后，等待被叫用户摘机的最大等待时间。有关 AutoDial 任务的执行过程请参见第 1 章中下列部分内容： ✧ 模拟中继线通道：“模拟中继线通道的状态转移”； ✧ SS1 通道：“中国 1 号信令的状态转移”中的T6 定时器； ✧ ISUP通道：“ISUP通道的状态迁移”中的T4 定时器； ✧ TUP通道：“TUP通道的状态迁移”中的T4 定时器。

配置项	**RingToPending**
节	**[TUP] / [ISUP]**
书写格式	RingToPending=m
取值范围	m=0：响应对端的拆线消息，开始拆除连接； m=1：不响应对端交换机的拆线消息，通道迁移到“挂起”状态，并通过设置挂起原因为 PEND_RemoteHangupOnRinging，由应用程序自行决定何时拆除连接。 缺省值为 0。
功能描述	来话呼叫过程中，当本端处于“振铃”状态时，如果对端交换机放弃呼叫，会向本端发送拆线消息。当驱动程序收到此拆线消息后，本配置项用于控制通道迁移到下一个状态的路径。有关本配置项的更多信息请参见第 1 章中通道的状态转移部分内容。
注意事项	● 本配置项只适用于 TUP 通道和 ISUP 通道 ● TUP 通道的配置项使用[TUP]配置节，ISUP 通道的配置项使用[ISUP]配置节

配置项	**AutoClearCallerIdBufOnHangup**
节	**[SystemConfig]**
书写格式	AutoClearCallerIdBufOnHangup=b
取值范围	b=0：否（缺省值） b=1：是
功能描述	呼叫结束时，本配置项决定是否由驱动程序自动清除保存主叫方号码缓冲区和扩展缓冲区。如果是 ISUP 通道或 TUP 通道，保存被叫方号码和第 1 被叫号码的缓冲区也将被清空。
注意事项	本配置项要求驱动程序的版本为 2.1 或更高

配置项	**AutomaticAisGeneration**
节	**[BoardId=x]**
书写格式	AutomaticAisGeneration[n]=b
取值范围	n：逻辑 Pcm 编号，从 0 开始编号； b=0：否；（缺省值） b=1：是。
功能描述	当输入信号中断时，是否启用自动生成 AIS 消息。
注意事项	● 本配置项仅用于 8E1/16E1 的链路 ● 对第 4m 个逻辑 PCM 设置本配置项时，系统同时会自动设置第（4*m+1）个、第（4*m+2）个、第（4*m+3）个逻辑 PCM 的 AutomaticAisGeneration（其中 m 是使逻辑 Pcm 索引有意义的自然数 ● 本配置项要求驱动程序的版本为 Ver4.7.3.1 或更高

配置项	**AutomaticRaiGereration**
节	**[BoardId=x]**
书写格式	AutomaticRaiGereration =b
取值范围	b=0：否（缺省值） b=1：是
功能描述	当输入信号丢失时，是否启用自动生成 RAI 消息。
注意事项	● 本配置项仅用于 8E1/16E1 的链路 ● 对第 4m 个逻辑 PCM 设置本配置项时，系统同时会自动设置第（4*m+1）个、第（4*m+2）个、第（4*m+3）个逻辑 PCM 的 AutomaticAisGeneration（其中 m 是使逻辑 Pcm 索引有意义的自然数 ● 本配置项要求驱动程序的版本为 Ver4.7.3.1 或更高

配置项	**IgnoreBlockInGra**
节	**[TUP]**
书写格式	IgnoreBlockInGra =b
取值范围	b=0：否（缺省值） b=1：是
功能描述	是否忽略 GRA 中出现的状态指示域
注意事项	本配置项仅在 TUP 接续中有意义

3.1.2.6 设置数字线路上的语音编码格式

配置项	**DefaultVoiceFormat**
节	**[BoardId=x]**
书写格式	DefaultVoiceFormat=m
取值范围	m=6：A-Law； m=7：μ_Law。
功能描述	设置数字线路（包括 DST 系列板卡所连接的数字用户线和 SHD/DTP 系列板卡所连接的数字中继线）上 B 信道语音数据的编码格式。 北美地区和日本通常使用 μ_Law，欧洲和中国使用 A-Law。B 信道语音数据的编码格式可以通过查阅交换机或电话机的相关文档来确定，也可以通过实验方法来确定。
注意事项	● 本配置项只适用于 DST、SHD、DTP 系列板卡； ● 如果本配置项的设定值与实际的 B 信道语音数据编码格式不一致，采集到的语音数据会出现杂音。

3.1.2.7 设置信号音检测器

3.1.2.7.1 设置数字卡的信号音检测开关

配置项	**DefaultToneCheckState**
节	**[BoardId=x]**
书写格式	**DefaultToneCheckState** =n
取值范围	n=0：关闭（缺省值） n=1：开启信号音检测
功能描述	设置数字卡的信号音检测开关

3.1.2.7.2 设置侦测的信号音参数

配置项	**DefaultDetectFrequence** **DefaultDetectBandWidth**

节	**[BoardId=x]**
书写格式	DefaultDetectFrequence=e,e,e,e DefaultDetectBandWidth=f,f,f,f,f
取值范围	e：侦测的信号音频率，缺省值为 450； f：对应四个频率的带宽和峰值频率的带宽，缺省值均为 80。
功能描述	DefaultDetectFrequence 设置侦测的四个频率值； DefaultDetectBandWidth 设置对应四个频率的带宽和峰值频率的带宽。

3.1.2.7.3 设置噪声滤波器的工作参数

配置项	**MinimumVoiceDetermineEnergy**
节	**[SystemConfig]**
书写格式	MinimumVoiceDetermineEnergy=e
取值范围	e：e≥700，缺省值为 100000。本参数的单位与DB的换算方法请参见第 1 章中“信号音检测器”节中的“FFT模块”部分
功能描述	设置线路上的噪音判别门限，详细信息请参见函数SsmSetMinVocDtrEnergy的说明

3.1.2.7.4 设置脉宽滤波器的工作参数 T

配置项	**ToneHighFilterPoint** **ToneLowFilterPoint**
节	**[SystemConfig]**
书写格式	ToneHighFilterPoint=t_H ToneLowFilterPoint=t_L
取值范围	t_H：高电平的最小保持时间，单位为 16 毫秒，t_H >0，取值范围：缺省值为 4 t_L：低电平的最小保持时间，单位为 16 毫秒，t_L >0，取值范围：缺省值为 10
功能描述	ToneHighFilterPoint 设置信号音的最小保持时间。如果来话信号中出现了信号音的频率特征，但其保持时间小于本参数的设定值，驱动程序认为是干扰信号； ToneLowFilterPoint 设置确认信号音消失的最小保持时间。在已经判定了来话信号中出现信号音后，如果该频率特征消失的时间大于本参数的设定值，驱动程序才确认消失。

3.1.2.7.5 设置第 1 呼叫进程音检测器

3.1.2.7.5.1 设置频率检测器的工作参数

配置项	**TonePara**
节	**[SystemConfig]**
书写格式	TonePara =f1,b1,f2,b2,r
取值范围	f1：　第 1 个中心频率，单位为 Hz，取值范围：300~3400，缺省值为 450 b1：　第 1 个中心频率的带宽，单位为 Hz，取值范围：40~120，缺省值为 80 f2：　第 2 个中心频率，单位为 Hz，取值范围：300~3400，缺省值为 0 b2：　第 2 个中心频率的带宽，单位为 Hz，取值范围：40~120，缺省值为 0 r：　判定门限，单位：%，取值范围：15~80，缺省值为 50 更多信息请参见函数SsmSetTonePara的说明。
功能描述	设置呼叫进程音检测器中频率检测器的工作参数，详细内容请参见第 1 章中呼叫进程音检测器部分

3.1.2.7.5.2 设置拨号音检测器的工作参数

配置项	**IsDialToneDetermineTime**
节	**[SystemConfig]**
书写格式	IsDialToneDetermineTime=t
取值范围	t：最小保持时间，单位为毫秒，t≥1200，缺省值为 1500
功能描述	设置第 1 呼叫进程音检测器的拨号音检测参数。详细内容请参见函数SsmSetIsDialToneDtrTime的说明

3.1.2.7.5.3 设置回铃音检测器的工作参数

配置项	**RingEchoTonePara**
节	**[SystemConfig]**
书写格式	RingEchoTonePara=t_H,t_L
取值范围	t_H：高电平保持时间，取值范围：300≤t_H≤2500，单位为毫秒，缺省值为 1000 t_L：高电平保持时间，取值范围：800≤t_L≤6000，单位为毫秒，缺省值为 14000
功能描述	设置回铃音检测器的高电平保持时间和低电平保持时间。详细内容请参见函数 SsmSetRingEchoTonePara的说明

3.1.2.7.5.4 设置忙音检测器的工作参数

配置项	**BusyTonePeriod** **IsBusyToneDetermineCount**
节	**[SystemConfig]**
书写格式	格式 1：BusyTonePeriod=p_1 格式 2：BusyTonePeriod=p_1,p_2 格式 3：BusyTonePeriod=p_1,p_2,p_3 格式 4：BusyTonePeriod=p_1,p_2,p_3,p_4 IsBusyToneDetermineCount=n
取值范围	p_1,p_2,p_3,p_4：忙音周期，单位为毫秒，取值范围：200≤p_i≤2000，缺省值 700ms。 n：　　忙音周期的最小持续个数，1≤n≤10，缺省值为 2
功能描述	设置第 1呼叫进程音检测器中忙音检测器的工作参数。详细内容请参见第 1 章中忙音检测器部分。 BusyTonePeriod设置忙音周期，最多可以设置 4 个，中间用“,”隔开。格式 1、格式 2、格式 3、格式 4 分别适用于设置 1~4 个周期的情况，详细内容请参见函数SsmSetBusyTonePeriodEx的说明； IsBusyToneDetermineCount 设置忙音周期的最小持续个数。
使用范例	BusyTonePeriod=700　　//检测 1 种忙音，周期为 700 毫秒 BusyTonePeriod=700,1400　//检测 2 种忙音，周期分别为 700 毫秒和 1400 毫秒

3.1.2.7.5.5 设置自定义信号音检测器的工作参数

配置项	**AppointedToneAnalyzerSwitch**
节	**[SystemConfig]**
书写格式	AppointedToneAnalyzerSwitch=m
取值范围	m=0：关闭； m=1，开启，检测连续性信号音，此时配置项IsAppointedToneDetermineTime有效； m=2：开启，检测周期性信号音，此时配置项AppointedTonePara和IsAppointedToneDetermineCount 有效
功能描述	设置第 1 呼叫进程音检测器的自定义信号音检测器的工作状态。详细内容请参见第 1 章中自定义信号音检测器部分。

配置项	**IsAppointedToneDetermineTime**
节	**[SystemConfig]**
书写格式	IsAppointedToneDetermineTime=t
取值范围	t≥16：最小保持时间，单位为毫秒，缺省值为 80。
功能描述	设置自定义的连续性信号音的最小保持时间，适用于第 1 呼叫进程音检测器的自定义信号音检测器。详细内容请参见第 1 章中自定义信号音检测器部分。
注意事项	● 只有当配置项AppointedToneAnalyzerSwitch设置为 1 时本配置项才有效； ● 如果本配置项的设定值小于配置项IsDialToneDetermineTime的设置值，会导致无法正常检测拨号音；如果本配置项的设定值大于IsDialToneDetermineTime的配置值，会导致无法检测自定义的连续性信号音。

配置项	**AppointedTonePara** **IsAppointedToneDetermineCount**
节	**[SystemConfig]**

书写格式	AppointedTonePara=t_H,t_L IsAppointedToneDetermineCount=n
取值范围	t_H：高电平保持时间，16≤t_H≤6000，单位为毫秒，缺省值为 2000 t_L：低电平保持时间，16≤t_L≤6000，单位为毫秒，缺省值为 2000 n：自定义周期性信号音的最小持续个数，1≤n≤10，缺省值为 2
功能描述	设置第 1呼叫进程音检测器中自定义信号音检测器的周期性信号音工作参数。 AppointedTonePara 设置高电平保持时间和低电平保持时间，IsAppointedToneDetermineCount 设置最小计数周期
注意事项	只有当配置项AppointedToneAnalyzerSwitch的设定值为 2 时，本配置项才有效。

3.1.2.7.6 设置第 2 呼叫进程音检测器

3.1.2.7.6.1 设置工作状态

配置项	**Enable2ndToneAnalyzer**
节	**[SystemConfig]**
书写格式	Enable2ndToneAnalyzer=b
取值范围	b=1：开启 b=0：关闭
功能描述	设置第 2 呼叫进程音检测器的工作状态。详细内容请参见函数SsmStart2ndToneAnalyzer的说明。

配置项	**Check2ndToneOnAutoDial**
节	**[SystemConfig]**
书写格式	Check2ndToneOnAutoDial=b
取值范围	b=1：是 b=0：否（缺省值）
功能描述	设置第 2 呼叫进程音检测器的检测结果是否影响模拟中继线通道的状态迁移。更多信息请参见第 1 章中“呼叫进程音检测器”部分内容。

3.1.2.7.6.2 设置频率检测器的工作参数

配置项	**2ndTonePara**
节	**[SystemConfig]**
书写格式	TonePara =f_1,b_1,f_2,b_2,r
取值范围	f_1：第 1 个中心频率，单位为 Hz，取值范围：300~3400，缺省值为 450 b_1：第 1 个中心频率的带宽，单位为 Hz，取值范围：40~120，缺省值为 80 f_2：第 2 个中心频率，单位为 Hz，取值范围：300~3400，缺省值为 0 b_2：第 2 个中心频率的带宽，单位为 Hz，取值范围：40~120，缺省值为 0 r： 判定门限，单位：%，取值范围：15~80，缺省值为 50 更多信息请参见函数SsmSet2ndTonePara的说明。
功能描述	设置第 2 呼叫进程音检测器中频率检测器的工作参数，详细内容请参见第 1 章中呼叫进程音检测器部分。

3.1.2.7.6.3 设置拨号音检测器的工作参数

配置项	**2ndIsDialToneDetermineTime**
节	**[SystemConfig]**
书写格式	2ndIsDialToneDetermineTime=t
取值范围	t：拨号音的最小保持时间，单位为毫秒，n≥1200，缺省值为 1500
功能描述	设置第 2 呼叫进程音检测器的拨号音检测参数。详细内容请参见函数SsmSet2ndIsDialToneDtrTime的说明。

3.1.2.7.6.4 设置回铃音检测器的工作参数

配置项	**2ndRingEchoTonePara**

节	**[SystemConfig]**
书写格式	2ndRingEchoTonePara=t_H,t_L
取值范围	t_H：高电平保持时间，300≤t_H≤2500，单位为毫秒，缺省值为 1000 t_L：低电平保持时间，800≤t_L≤6000，单位为毫秒，缺省值为 14000
功能描述	设置第 2 呼叫进程音检测器的回铃音检测器的高电平保持时间和低电平保持时间。详细内容请参见函数SsmSet2ndRingEchoTonePara的说明。

3.1.2.7.6.5 设置忙音检测器的工作参数

配置项	**2ndBusyTonePeriod** **2ndIsBusyToneDetermineCount**
节	**[SystemConfig]**
书写格式	格式 1：2ndBusyTonePeriod=p_1 格式 2：2ndBusyTonePeriod=p_1,p_2 格式 3：2ndBusyTonePeriod=p_1,p_2,p_3 格式 4：2ndBusyTonePeriod=p_1,p_2,p_3,p_4 2ndIsBusyToneDetermineCount=n
取值范围	p_1,p_2,p_3,p_4：忙音周期，单位为毫秒，取值范围：200≤p_i≤2000，缺省值 700ms。 n：　　忙音周期的最小持续个数，1≤n≤10，缺省值为 2
功能描述	设置第 2呼叫进程音检测器中忙音检测器的工作参数。 2ndBusyTonePeriod设置忙音周期，最多可以设置 4 个，中间用“,”隔开。格式 1、格式 2、格式 3、格式 4 分别适用于设置 1~4 个周期的情况，详细内容请参见函数SsmSetBusyTonePeriodEx的说明； 2ndIsBusyToneDetermineCount 设置忙音周期的最小持续个数。
使用范例	2ndBusyTonePeriod=700　　//检测 1 种忙音，周期为 700 毫秒 2ndBusyTonePeriod=700,1400　//检测 2 种忙音，周期分别为 700 毫秒和 1400 毫秒

配置项	**MaxBsTnOffTime**
节	**[SystemConfig]**
书写格式	MaxBsTnOffTime=t
取值范围	t≥400：单位为毫秒，缺省值为 2000 毫秒
功能描述	设置忙音检测的保护时间。详细内容请参见第 1 章中忙音检测器部分。
注意事项	本配置项对第 2 呼叫进程音检测器也起作用

3.1.2.7.6.6 设置自定义信号音检测器的工作参数

配置项	**2ndAppointedToneAnalyzerSwitch**
节	**[SystemConfig]**
书写格式	2ndAppointedToneAnalyzerSwitch=m
取值范围	m=0：关闭； m=1，开启，检测连续性信号音； m=2：开启，检测周期性信号音。
功能描述	设置第 2 呼叫进程音检测器的自定义信号音检测器的工作状态。详细内容请参见第 1 章中自定义信号音检测器部分。

配置项	**2ndIsAppointedToneDetermineTime**
节	**[SystemConfig]**
书写格式	2ndIsAppointedToneDetermineTime=t
取值范围	t≥16：最小保持时间，单位为毫秒，缺省值为 80。
功能描述	设置自定义的连续性信号音的最小保持时间，适用于第 2 呼叫进程音检测器的自定义信号音检测器。详细内容请参见第 1 章中自定义信号音检测器部分。
注意事项	● 只有当配置项2ndAppointedToneAnalyzerSwitch设置 1 时本配置项才有效； ● 如果本配置项的设定值小于配置项2ndIsDialToneDetermineTime的设置值，会导致无法正常检测拨号音；如果本配置项的设定值大于2ndIsDialToneDetermineTime的配置值，会导致无法检测自定义的连续性信号音。

配置项	**2ndAppointedTonePara** **2ndIsAppointedToneDetermineCount**
节	**[SystemConfig]**
书写格式	2ndAppointedTonePara=t_H,t_L 2ndIsAppointedToneDetermineCount=n
取值范围	t_H：高电平保持时间，16≤t_H≤6000，单位为毫秒，缺省值为 2000 t_L：低电平保持时间，16≤t_L≤6000，单位为毫秒，缺省值为 2000 n：自定义周期性信号音的最小持续个数，1≤n≤10，缺省值为 2
功能描述	设置第 2呼叫进程音检测器中自定义信号音检测器的周期性信号音的工作参数。 2ndAppointedTonePara 设置高电平保持时间和低电平保持时间，2ndIsAppointedToneDetermineCount 设置最小计数周期。
注意事项	只有当配置项2ndAppointedToneAnalyzerSwitch的设定值为 2 时，本配置项才有效。

3.1.2.7.7 设置传真进程音检测器

配置项	**VoiceFreqF1Para** **VoiceFreqF2Para**
节	**[SystemConfig]**
书写格式	VoiceFreqF1Para=f_1,b_1,r_1,t_1 VoiceFreqF2Para=f_2,b_2,r_2,t_2
取值范围	f_1，f_2：中心频率，单位为 Hz，取值范围：300~3400，f_1 的缺省值为 1100，f_2 的缺省值为 2100。 b_1，b_2：带宽，单位为 Hz，取值范围：40~120，缺省值均为 80。 r_1，r_2：带内能量占全频能量的百分比的最小门限值，单位为%，取值范围：15 ~ 80，缺省值均为 50。 t_1，t_2：最小保持时间，单位：毫秒，取值范围：≥64，缺省值均为 250。
功能描述	设置传真进程音检测器的工作参数。VoiceFreqF1Para 设置第 1 传真进程音检测器，VoiceFreqF2Para 设置第 2 传真进程音检测器。 详细内容请参见函数SsmSetVoiceFxPara的说明。

3.1.2.7.8 设置简易忙音检测器

3.1.2.7.8.1 设置工作状态

配置项	**ToneAnalyzeAtRcvFsk**
节	**[BoardId=x]**
书写格式	ToneAnalyzeAtRcvFsk=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	当FSK数据接收器正在工作时，是否开启简易忙音检测器。更多信息请参见第 1 章中“简易忙音检测器”部分内容。
注意事项	本配置项通常只用于 SHT 系列板卡的模拟中继线通道。

3.1.2.7.8.2 设置噪声滤波器的工作参数

配置项	**TAARFDetermineEnergy**
节	**[BoardId=x]**
书写格式	TAARFDetermineEnergy=e
取值范围	e≥700，缺省值为 100000。 本参数的单位与DB的换算方法请参见第 1 章中“信号音检测器”节中的“FFT模块”部分。
功能描述	设置线路上的噪音判别门限，详细内容请参见第 1 章中简易忙音检测器部分。

3.1.2.7.9 设置回铃音检测方式

3.1.2.7.9.1 设置回铃音检测方式

配置项	IsEchoQuickDetect
节	[BoardId=x]
书写格式	IsEchoQuickDetect=b
取值范围	b=0：必须检测到完整的回铃音，并且回铃音从低电平开始计算（缺省值）； b=1：根据 EchoOnTolerance、EchoOffTolerance 设置的误差检测到回铃音即可，并且回铃音可从低电平计算，也可从高电平开始计算。
功能描述	设置回铃音检测方式。

3.1.2.7.9.2 设置检测回铃音高电平时允许的误差

配置项	EchoOnTolerance
节	[BoardId=x]
书写格式	EchoOnTolerance=e
取值范围	0≤e≤50，缺省值为 40。
功能描述	设置检测回铃音高电平时允许的误差。若设置为 40，则允许 40%的误差，即 1 秒高电平检测到 600 毫秒时就认为是回铃音高电平。
注意事项	与配置项 IsEchoQuickDetect、EchoOffTolerance 配合使用。

3.1.2.7.9.3 设置检测回铃音低电平时允许的误差

配置项	EchoOffTolerance
节	[BoardId=x]
书写格式	EchoOffTolerance =e
取值范围	0≤e≤50，缺省值为 50。
功能描述	设置检测回铃音低电平时允许的误差。若设置为 50，则允许 50%的误差，即 4 秒低电平检测到 2 秒时就认为是回铃音低电平。
注意事项	与配置项 IsEchoQuickDetect、EchoOnTolerance 配合使用。

3.1.2.8 设置信号音发生器(CTI 系列)

配置项	DefaultSendToneFrequence DefaultSendToneVolume
节	[BoardId=x]
书写格式	DefaultSendToneFrequence=f_1,f_2 DefaultSendToneVolume=v_1,v_2
取值范围	f_1：第 1 音频的频率，单位为 Hz，缺省值为 450； f_2：第 2 音频的频率，单位为 Hz。f_2 为 0 表示单音频信号音，大于 0 表示双音频信号音。缺省值为 0； v_1：第 1 音频的音量，取值范围：-7≤v_1≤+6，-7 表示关闭该频率，大于 0 表示音量提升，小于 0 表示音量衰减，设置值乘以 3 即为分贝值，缺省值为 0； v_2：第 2 音频的音量，取值范围及含义同 v_1，缺省值为-7。
功能描述	设置信号音发生器的工作参数。DefaultSendToneFrequence 设置音频信号的频率，DefaultSendToneVolume 设置音量。
使用范例	DefaultSendToneFrequence=450, 600　　//使用双音频信号音 DefaultSendToneVolume=0,0

3.1.2.9 设置自动语音识别

配置项	LoadRecognize voiceFormat
节	[SystemConfig]

书写格式	LoadRecognize=n voiceFormat＝m
取值范围	n ＝0，不支持语音识别功能（缺省值） ＝1，支持语音识别功能； m＝1，A-Law 8k 单声道（缺省值） ＝0，μ-Law 8k 单声道.。
功能描述	LoadRecognize 设置是否支持语音识别； voiceFormat 设置语音识别引擎输入数据编码格式。

3.1.2.10 设置电话会议参数(CTI 系列)

配置项	**ClearInVoiceOnRxDtmf**
节	**[BoardId=x]**
书写格式	格式 1：ClearInVoiceOnRxDtmf=n,n,n,n 格式 2：ClearInVoiceOnRxDtmf=n,n,n,n,n,n,n,n 格式 3：ClearInVoiceOnRxDtmf=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4：ClearInVoiceOnRxDtmf=n
取值范围	n=0：启用（缺省值）； n=1：关闭。
功能描述	设置是否启用DTMF钳位功能。更多信息请参见第 1 章中“分布式电话会议”部分内容。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于 SHD 系列板卡； ● 函数函数SsmSetFlag（携带F_ClearInVoiceOnRcvDtmf参数）也可实现相同功能。

配置项	**ConfMaxGroup**
节	**[SystemConfig]**
书写格式	ConfMaxGroup=n
取值范围	n：n≥0（缺省值为 64）
功能描述	设置可以创建的会议室的最大个数。

配置项	**ConfDefaultMaxGroupMember**
节	**[SystemConfig]**
书写格式	ConfDefaultMaxGroupMember=n
取值范围	n：n≥0（缺省值为 64）
功能描述	设置可以加入会议室的会议通道的最大个数。

配置项	**ConfDefaultMaxGroupSpeaker**
节	**[SystemConfig]**
书写格式	ConfDefaultMaxGroupSpeaker=n
取值范围	n：会议通道的个数，缺省值为 64。
功能描述	设置会议室中具有发言权的会议通道的最大个数。主持人、主席、动态发言模式均具有发言权模式，背景音乐模式是否具有发言权取决于配置项BackgroundVoicePriority的设置。

配置项	**ConfDefaultMaxGroupSpeaking**
节	**[SystemConfig]**
书写格式	ConfDefaultMaxGroupSpeaking=n
取值范围	n≤6，缺省值为 6。
功能描述	设置会议室中允许同时发言的最大通道数。

配置项	**ConfJoinedbyEnergy**
节	**[SystemConfig]**
书写格式	ConfJoinedbyEnergy=n

取值范围	n=0：无抢占； n=2：支持抢占式发言(缺省值)。
功能描述	设置电话会议中动态发言时，是否支持抢占式发言。

配置项	**ConfMaxListener**
节	**[SystemConfig]**
书写格式	ConfMaxListener =n
取值范围	n：会议通道数，缺省值为 480。
功能描述	设置会议室中允许加入的最大听众数。

配置项	**ConfDefaultMaxSilenceTime**
节	**[SystemConfig]**
书写格式	ConfDefaultMaxSilenceTime=t
取值范围	t：保持时间，单位为秒，缺省值为 5 秒。
功能描述	设置判断通道是否保持静默的最小保持时间。更多信息请参见第 1 章中“分布式电话会议”部分内容。

配置项	**BackgroundVoicePriority**
节	**[SystemConfig]**
书写格式	BackgroundVoicePriority=n
取值范围	n=1：高于主持人，具有专用的会议混音器； n=2：低于主持人，但高于主席（缺省值）； n=3：低于主席，但高于动态发言； n=4：低于动态发言。
功能描述	设置播放背景音乐的通道的发言模式的调度优先级。更多信息请参见第 1 章中“分布式电话会议”部分内容。

配置项	**PlayVoiceIsListen**
节	**[SystemConfig]**
书写格式	PlayVoiceIsListen =n
取值范围	n=0：此通道只说不听； n=1：此通道可说可听。
功能描述	设置电话会议中用于播放背景音的通道是否可听。

3.1.2.11 设置放音操作

3.1.2.11.1 设置播放任务的终止条件

配置项	**DefaultDtmfStopPlay** **DtmfStopPlayCharSet**
节	**[BoardId=x]**
书写格式	格式 1：DefaultDtmfStopPlay=b,b,b,b DtmfStopPlayCharSet={s,s,s,s} 格式 2：DefaultDtmfStopPlay=b,b,b,b,b,b,b,b DtmfStopPlayCharSet={s,s,s,s,s,s,s,s} 格式 3：DefaultDtmfStopPlay=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b DtmfStopPlayCharSet={s,s,s,s,s,s,s,s,s,s,s,s,s,s,s,s} 格式 4：DefaultDtmfStopPlay=b DtmfStopPlayCharSet=s
取值范围	b=0：否（缺省值），此时 DtmfStopPlayCharSet 无效； b=1：是，此时 DtmfStopPlayCharSet 有效。 s={0123456789*#abcd }，可以任意个数的 DTMF 字符集，缺省值为全集。
功能描述	DefaultDtmfStopPlay设置放音任务是否因DTMF检测器检测到DTMF字符而终止，如果设置为是，DtmfStopPlayCharSet设置DTMF字符集。

注意事项	格式 1 适用于 SHT 系列的 USB 语音盒； 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列； 格式 4 适用于 SHD 系列。
使用范例	DefaultDtmfStopPlay=1 DtmfStopPlayCharSet={#}　　// SHD 系列：检测到#就停止放音任务，其余字符不起作用

配置项	**HangupStopPlay**
节	**[BoardId=x]**
书写格式	格式 1：HangupStopPlay=b,b,b,b 格式 2：HangupStopPlay=b,b,b,b,b,b,b,b 格式 3：HangupStopPlay=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：HangupStopPlay=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置放音任务是否因驱动程序状态机检测到对端挂机而终止。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒； 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列； 格式 4 适用于 SHD 系列板卡。

配置项	**DefaultBargeInStopPlay**
节	**[BoardId=x]**
书写格式	格式 1：DefaultBargeInStopPlay=b,b,b,b 格式 2：DefaultBargeInStopPlay=b,b,b,b,b,b,b,b 格式 3：DefaultBargeInStopPlay=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：DefaultBargeInStopPlay=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置放音任务是否因Barge in检测器检测到语音活动而终止。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒； 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列； 格式 4 适用于 SHD 系列板卡。

3.1.2.11.2 设置驱动程序内部的放音缓冲区

配置项	**PlayBufSize**
节	**[SystemConfig]**
书写格式	PlayBufSize=k
取值范围	k：放音缓冲区的长度，单位为字节，必须为 512 的整数倍，缺省值为 32768 字节。
功能描述	设置放音缓冲区大小。

3.1.2.11.3 设置文件序列播放任务的参数

配置项	**MaxPlayFileList**
节	**[SystemConfig]**
书写格式	MaxPlayFileList=n
取值范围	0≤n≤65535，缺省值为 256。
功能描述	设置允许应用程序调用SsmAddToFileList向驱动程序提交的语音文件数量的上限。

3.1.2.11.4 设置预加载文件播放任务的参数

配置项	**MaxPlayIndexList**
节	**[SystemConfig]**
书写格式	MaxPlayIndexList=n
取值范围	0≤n≤65535，缺省值为 256。
功能描述	设置调用一次SsmPlayIndexList或SsmPlayIndexString函数所允许的语音片段个数的上限。

3.1.2.11.5 设置内存序列播放任务的参数

配置项	**MaxPlayMemList**
节	**[SystemConfig]**
书写格式	MaxPlayMemList=n
取值范围	0≤n≤65535，缺省值为 256。
功能描述	设置允许应用程序调用SsmAddToPlayMemList向驱动程序提交的缓冲区数量的上限。

3.1.2.11.6 设置文件播放任务的参数

配置项	**FastPlayTime**
节	**[SystemConfig]**
书写格式	FastPlayTime=t
取值范围	t 表示一次快进或快退操作对应的步长，单位为毫秒，缺省值为 1000。
功能描述	设置SsmFastFwdPlay和SsmFastBwdPlay函数的步长。

3.1.2.11.7 设置播放操作的音量

配置项	**DefaultPlayVolume**
节	**[BoardId=x]**
书写格式	格式 1：DefaultPlayVolume=v,v,v,v 格式 2：DefaultPlayVolume=v,v,v,v,v,v,v,v 格式 3：DefaultPlayVolume=v,v,v,v,v,v,v,v,v,v,v,v,v,v,v,v 格式 4：DefaultPlayVolume=v
取值范围	-6≤v≤+6，缺省值为 0。
功能描述	设置 SHD/SHT 系列板卡中音量调节器 A1 的增益。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒； 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡、DST 系列板卡； 格式 4 适用于 SHD 系列板卡、DTP 系列卡。

3.1.2.11.8 设置播放操作的编码格式

配置项	**DefaultPlayFormat**
节	**[BoardId=x]**
书写格式	格式 1：DefaultPlayVolume=v,v,v,v 格式 2：DefaultPlayVolume=v,v,v,v,v,v,v,v 格式 3：DefaultPlayVolume=v,v,v,v,v,v,v,v,v,v,v,v,v,v,v,v 格式 4：DefaultPlayVolume=v
取值范围	V=6，A 率（缺省值）； V=7，μ 率； V=17，ADPCM。
功能描述	设置通道的缺省放音数据编码格式。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒； 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡、DST 系列板卡； 格式 4 适用于 SHD 系列板卡、DTP 系列卡。

3.1.2.11.9 设置播放操作的其它参数

配置项	**DefaultPausePlayOnRxDtmf**
节	**[BoardId=x]**

书写格式	格式 1：DefaultPausePlayOnRxDtmf=b,b,b,b 格式 2：DefaultPausePlayObRxDtmf=b,b,b,b,b,b,b,b 格式 3：DefaultPausePlayObRxDtmf=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：DefaultPausePlayObRxDtmf=b 格式 5：DefaultPausePlayObRxDtmf=[0,29]:b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	当 DTMF 检测器在来话信号中检测到 DTMF 信号时，如果通道上正在执行语音播放的任务，为了保证 DTMF 检测器的准确性，驱动程序需要暂时停止语音的播放，待 DTMF 信号消失后，再自动恢复语音的播放。本配置项设置是否需要启动此项功能。 一般而言，录音通道应该设置为 0，其余类型通道设置为 1。
注意事项	● 格式 1 适用于 SHT 系列、ATP 系列的 USB 语音盒/录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 4 适用于 SHD 系列板卡； ● 格式 5 适用于 30 通道规格的远端坐席接入模块。

3.1.2.12 设置录音操作

3.1.2.12.1 设置录音任务的终止条件

配置项	**DtmfStopRecCharSet**
节	**[BoardId=x]**
书写格式	格式 1：DtmfStopRecCharSet={s,s,s,s} 格式 2：DtmfStopRecCharSet={s,s,s,s,s,s,s,s} 格式 3：DtmfStopRecCharSet={s,s,s,s,s,s,s,s,s,s,s,s,s,s,s,s} 格式 4：DtmfStopRecCharSet=s
取值范围	s 为 DTMF 字符集，书写形式为{***}，其中*可以是“0123456789*#abcd”中的任何一个，个数可根据实际需要设置。s 为空集表示关闭此项功能，非空表示开启此项功能。缺省值为空集。
功能描述	设置录音任务是否因DTMF检测器检测到DTMF字符而终止
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒； 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡； 格式 4 适用于 SHD 系列、DTP 系列板卡。
使用范例	DtmfStopRecCharSet={0123456789*#abcd } //SHD 系列：检测到任意字符就停止录音任务

配置项	**HangupStopRec**
节	**[BoardId=x]**
书写格式	格式 1：HangupStopRec=b,b,b,b 格式 2：HangupStopRec=b,b,b,b,b,b,b,b 格式 3：HangupStopRec=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：HangupStopRec=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置录音任务是否因驱动程序状态机检测到对端挂机而终止。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒； 格式 2、格式 3 分别适用于 8 通道、16 规格的 SHT 系列和 ATP 系列板卡； 格式 4 适用于 SHD 系列板卡。

3.1.2.12.2 设置录音信号源

配置项	**DefaultRecordVolume** **DefaultRecordMixerVolume**
节	**[BoardId=x]**

书写格式	格式 1：DefaultRecordVolume=v_3,v_3,v_3,v_3 DefaultRecordMixerVolume=v_2,v_2,v_2,v_2 格式 2：DefaultRecordVolume=v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3 DefaultRecordMixerVolume=v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2 格式 3：DefaultRecordVolume= v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3,v_3 DefaultRecordMixerVolume=v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2,v_2 格式 4：DefaultRecordVolume=v_3 DefaultRecordMixerVolume=v_2
取值范围	v_3,v_2：音量调节器的音量，$-7 \le v_3 \le +6$，$-7 \le v_2 \le +6$，-7 表示关闭，其余取值表示音量增益（乘以 3 即为分贝），大于 0 表示音量提升，小于 0 表示音量衰减。v_3 的缺省值为 0，v_2 的缺省值为-7。
功能描述	设置进入录音混音器 M3 的信号增益。DefaultRecordVolume 设置音量调节器 A3 的增益，DefaultRecordMixerVolume 设置音量调节器 A2 的增益。更多信息参见第 1 章中相应板卡的原理框图部分。
注意事项	式 1 适用于 SHT 系列的 USB 语音盒、ATP 系列的 USB 录音盒； 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列，DefaultRecordVolume 还适用于 DST 系列板卡； 格式 4 适用于 SHD 系列、DTP 系列板卡。
使用范例	DefaultRecordVolume=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 //提升 3DB

3.1.2.12.3 设置文件录音的截尾功能

配置项	**TruncateTailOnRecordToFile**
节	**[SystemConfig]**
书写格式	TruncateTailOnRecordToFile=t
取值范围	t=0：关闭此项功能（缺省值）； t>0：从文件尾部删除的语音数据长度，单位为毫秒。
功能描述	设置文件录音时的截尾时间长度。有关文件录音截尾功能的详细内容请参见函数 SsmSetTruncateTail的说明。
注意事项	要求 SynCTI Ver. 2.1 或更高

3.1.2.12.4 设置驱动程序内部的录音缓冲区

配置项	**RecordBufSize**
节	**[SystemConfig]**
书写格式	RecordBufSize=k
取值范围	k：缓冲区的长度，单位为字节，必须为 512 的整数倍，缺省值为 32768 字节。
功能描述	设置驱动程序内部为通道分配的录音缓冲区的大小。

3.1.2.12.5 设置录音文件为 wav 格式

配置项	**RecAsWavFormat**
节	**[SystemConfig]**
书写格式	RecAsWavFormat =K
取值范围	K=0：维持原状，即文件扩展名为 wav 或 mp3 的按照 wav 格式进行录音 K=1：录音文件按照 wav 格式进行录音
功能描述	设置录音文件为 wav 格式

3.1.2.12.6 设置 AGC 的工作参数

配置项	**AGCMAXGAIN** **AGCMINGAIN**
节	**[BoardId=x]**
书写格式	AGCMAXGAIN=G_{max} AGCMINGAIN=G_{min}

取值范围	G_{max}：1≤G_{max}≤255，缺省值为 255。当设为最大值 255 时，AGC 的最大增益为 24dB。这个参数用于控制 AGC 对小信号的最大放大倍数。参数越大，越能提升小信号的电平，但是背景噪声也将被放大。 G_{min}：1≤G_{min}≤255，缺省值为 6。当设为缺省值 6 时，AGC 的最小增益为-8.5dB。这个参数用于控制 AGC 对大信号的最大衰减。如果期望 AGC 控制器只有放大作用，没有衰减作用，则本参数应大于或等于 16。
功能描述	用于控制 DSP 增益系数的变化范围的参数。
注意事项	AGC 的原始增益系数为 16，对应的增益为 0dB(既不放大也不衰减)。当输出信号电平小于预期电平时，增益系数将上升，反之则下降。AGC 的实际电平增益为 20log（增益系数/16）dB。本配置项为高级配置项。大多数情况下，使用缺省值就可以满足系统要求。如果确实需要修改此配置项，请在本公司技术人员的指导下进行。

配置项	**AGCMAXLEVEL** **AGCMINLEVEL**
节	**[BoardId=x]**
书写格式	AGCMAXLEVEL=L_{max} AGCMINLEVEL=L_{min}
取值范围	L_{max}：AGCMINLEVEL≤L_{max}≤25600000，缺省值为 3200000。输出电平的上限，如果实际的输出电平大于 L_{max}，AGC 模块开始降低增益，直到到达 AGCMINGAIN 设定的最小增益。 L_{min}： 0≤L_{min}≤AGCMAXLEVEL，缺省值为 2560000。输出电平的下限，如果实际输出电平小于 L_{min}，并且保持时间一定时间后，AGC 模块开始增大增益，直到到达配置项 AGCMAXGAIN 设定的最大增益。
功能描述	用于控制 AGC 输出电平期望值的参数。
注意事项	AGC 通过调节增益，将输出信号尽可能控制在指定范围内。如果希望输出结果的电平尽可能平稳，则可以将两个参数设为接近的值。但最好不要设为相同的值，否则将导致增益频繁调整，使得在稳定的输入电平情况下，AGC 输出电平反而上下波动。电平值的满幅度=25600000，如果假定满幅度电平为 0dB，则设定电平= 10log（设定值/25600000）dB，缺省值 3200000 和 2560000 分别相当于-9dB 和-10dB。本配置项为高级配置项。大多数情况下，使用缺省值就可以满足系统要求。如果确实需要修改此配置项，请在本公司技术人员的指导下进行。

配置项	**AGCDOWNRATIO** **AGCUPRATIO** **AGCKEEPTIME**
节	**[BoardId=x]**
书写格式	AGCDOWNRATIO=R_{down} AGCUPRATIO=R_{up} AGCKEEPTIME=t
取值范围	R_{down}：0≤R_{down}≤100，缺省值为 35。自动衰减增益的步长，5 表示 5%。控制增益下降速度。通常情况下应该设定较快的下降速度，从而减少输出信号的碰顶失真。 R_{up}： 0≤R_{up}≤100，缺省值为 1。自动提升增益的步长，5 表示 5%。控制增益上升速度。通常情况下应该设定较慢的上升速度，从而减少输出信号的背景噪声。但设得过小将使小信号得不到及时放大。 t： 0≤t≤32768，缺省值为 0。保持时间，每个单位表示 8 毫秒。增益由下降转为上升时，可以保持一段时间再开始。保持时间=设定值*16ms。设定本参数，有助于减少话音间隔的背景噪声，但会延迟对小信号的提升。增益下降没有保持时间。
功能描述	用于控制 AGC 增益变化速度的参数。
注意事项	DOWN_TIME 和 UP_TIME 所设定的增益变化速度= 20log（1/(1-设定值)）dB/16ms。缺省的下降速度为 3.7dB/16ms，也就是大约经过 100ms 时间，增益可以从最大值降为 0dB。缺省的上升速度为 0.13dB/16ms，也就是大约经过 3 秒的时间，增益可以从 0dB 到达最大值。本配置项为高级配置项。大多数情况下，使用缺省值就可以满足系统要求。如果确实需要修改此配置项，请在本公司技术人员的指导下进行。

配置项	OpenRecEnAndPlayEnOnIdle
节	[BoardId=x]
书写格式	格式 1：OpenRecEnAndPlayEnOnIdle=b,b,b,b 格式 2：OpenRecEnAndPlayEnOnIdle=b,b,b,b,b,b,b,b 格式 3：OpenRecEnAndPlayEnOnIdle=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置模拟中继线通道在空闲状态下进行录音操作时是否开启 AGC 功能。
注意事项	● 本配置项只适用于 SHT 系列的模拟中继线通道； ● 本配置项要求 SynCTI 驱动程序的版本为 4.7.2.0 或更高。

3.1.2.13 设置录音/放音的共用参数

3.1.2.13.1 设置乒乓缓冲区的最小值

配置项	RecordAndPlayUseAsIP
节	[SystemConfig]
书写格式	RecordAndPlayUseAsIP=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置录音、放音操作是否支持微型缓冲区。更多信息请参见函数SsmPlayMemBlock、SsmRecordMemBlock的说明。

3.1.2.13.2 设置软件编码的 GSM 格式

配置项	GsmCodecEnable
节	[SystemConfig]
书写格式	GsmCodecEnable=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	驱动程序是否自动加载 macmcvt.dll 模块。macmcvt.dll 是 SynCTI 驱动程序提供的音频编码/解码器模块(ACM)，可以实现 GSM、MP3 的软件编码/解码，以及 G.729A 的软件解码。如果应用程序希望在 Synway 板卡上实现 GSM、MP3 或 G.729A 格式的录音或放音，而板卡上不具备相应的硬件编码器/解码器，可以通过设置本配置项，将外挂的 ACM 程序加载到 SynCTI 驱动程序中，从而使 SynCTI 驱动程序具有了软件方式的语音编码和解码能力。
注意事项	● 如果 b 设置为 1，必须保证 Windows 系统中已经安装了相应的音频编解码器(ACM)驱动程序； ● 本配置项只适用于 Windows 操作系统。

3.1.2.14 设置 DTMF 检测器

3.1.2.14.1 启动/关闭 DTMF 检测器

配置项	AlwaysEnableRxDtmf
节	[BoardId=x]
书写格式	AlwaysEnableRxDtmf=m
取值范围	m=0：DTMF检测器的工作状态由驱动程序根据通道状态的迁移情况自动控制，当通道进入通话状态时自动开启，离开通话状态时自动关闭，但应用程序也可以通过SsmEnableRxDtmf函数进行控制； m=1：DTMF 检测器一直处于开启状态。
功能描述	设置DTMF检测器的工作方式。

注意事项	本配置项只支持 SHD 系列板卡。

配置项	**RcvDtmfOnIdle**
节	**[BoardId=x]**
书写格式	格式 1：RcvDtmfOnIdle=b,b,b,b 格式 2：RcvDtmfOnIdle=b,b,b,b,b,b,b,b 格式 3：RcvDtmfOnIdle=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：RcvDtmfOnIdle=[0,29]:b
取值范围	b=0：关闭（缺省值）； b=1：开启。
功能描述	当坐席通道处于空闲状态时，设置DTMF检测器的工作状态。
注意事项	● 本配置项仅支持坐席通道 ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于 30 通道规格的远端坐席接入模块；

3.1.2.14.2 设置 DTMF 检测器的缓冲区大小

配置项	**RxDtmfBufSize**
节	**[SystemConfig]**
书写格式	RxDtmfBufSize=k
取值范围	k：缓冲区的大小，单位为字节，缺省值为 200。
功能描述	设置DTMF检测器的缓冲区大小。

3.1.2.14.3 设置 DTMF 滤波器的参数

配置项	**DualAndAllFreqEnScale**
节	**[BoardId=x]**
书写格式	格式 1：DualAndAllFreqEnScale=n,n,n,n 格式 2：DualAndAllFreqEnScale=n,n,n,n,n,n,n,n 格式 3：DualAndAllFreqEnScale=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4：DualAndAllFreqEnScale=m
取值范围	0≤n≤100，单位为%，缺省值为 12。 0≤m≤100，单位为%，默认为 32。
功能描述	设置DTMF信号的带内能量占全频能量的百分比判定门限。有关此配置项的详细说明请参见第 1 章中DTMF滤波器部分。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡，但下列型号的板卡要求 SynCTI Ver. 4.5.1.7 或更高： ✧ SHD-240A-CT/cPCI/SS7 ✧ SHD-240S-CT/cPCI/SS7 ✧ SHD-480A-CT/cPCI/SS7 ✧ SHD-480S-CT/cPCI/SS7 ● 函数SsmSetFlag（携带F_DualAndAllFreqEnScale参数）可以实现相同功能。

配置项	**HighAndLowFreqEnScale**
节	**[BoardId=x]**
书写格式	格式 1：HighAndLowFreqEnScale=n,n,n,n 格式 2：HighAndLowFreqEnScale=n,n,n,n,n,n,n,n 格式 3：HighAndLowFreqEnScale=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4：HighAndLowFreqEnScale=m
取值范围	0≤n≤100，单位为%，缺省值为 10。 0≤m≤100，单位为%，缺省值为 32。

功能描述	设置 DTMF 信号中，高频分量部分能量与低频分量部分能量之间的比例。只有当 DTMF 信号的高频分量部分能量与低频分量部分能量之间的比例大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡； ● 函数SsmSetFlag（携带F_HighAndLowFreqEnScale参数）可以实现相同功能。

配置项	**DtmfEnergy**
节	**[BoardId=x]**
书写格式	格式 1：DtmfEnergy=n,n,n,n 格式 2：DtmfEnergy=n,n,n,n,n,n,n,n 格式 3：DtmfEnergy=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4：DtmfEnergy=m
取值范围	n≥0，缺省值为 0。设置方法请参考配置项MinimumVoiceDetermineEnergy的说明。
功能描述	设置 DTMF 信号的带内能量的绝对值。只有当 DTMF 信号的带内能量大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡；

3.1.2.14.4设置 DTMF 脉宽滤波器的参数

配置项	**ReceiveDtmfSensitive**
节	**[BoardId=x]**
书写格式	格式 1：ReceiveDtmfSensitive=n,n,n,n 格式 2：ReceiveDtmfSensitive=n,n,n,n,n,n,n,n 格式 3：ReceiveDtmfSensitive=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4：ReceiveDtmfSensitive=n
取值范围	n 包含 8 个比特（记为 Bit7…Bit0），各比特含义如下： Bit3~Bit0：高电平最小保持时间，取值范围为 1~6，单位为 16 毫秒，缺省值为 3； Bit7~Bit4：低电平最小保持时间，取值范围为 0~6，单位为 16 毫秒，缺省值为 0。 n 采用 16 进制表示。例如， ReceiveDtmfSensitive=0x43 表示高电平、低电平的最小保持时间分别为 48、64 毫秒。
功能描述	设置DTMF信号的高电平、低电平的最小保持时间。更多信息请参见第 1 章中DTMF脉宽滤波器部分。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列系列板卡，但 SHD 系列的下列型号的板卡要求 SynCTI Ver. 4.5.1.7 或更高： ✧ SHD-240A-CT/cPCI/SS7 ✧ SHD-240S-CT/cPCI/SS7 ✧ SHD-480A-CT/cPCI/SS7 ✧ SHD-480S-CT/cPCI/SS7

配置项	**OmitABCD**
节	**[BoardId=x]**
书写格式	格式 1：OmitABCD=b,b,b,b 格式 2：OmitABCD=b,b,b,b,b,b,b,b 格式 3：OmitABCD=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：OmitABCD=b

取值范围	b=0：丢弃 ； b=1：不丢弃。 缺省值为 0
功能描述	设置是否丢弃DTMF检测器收到的'A'、'B'、'C'、'D'字符。有关此配置项的详细说明请参见第 1 章中DTMF脉宽滤波器部分。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT、ATP、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡；

配置项	**DtrmOnDtmfHighLevel**
节	**[BoardId=x]**
书写格式	格式 1：DtrmOnDtmfHighLevel=n,n,n,n 格式 2：DtrmOnDtmfHighLevel=n,n,n,n,n,n,n,n 格式 3：DtrmOnDtmfHighLevel=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4：DtrmOnDtmfHighLevel=n
取值范围	n=0：DTMF信号的低电平保持时间超过配置项ReceiveDtmfSensitive 指定的最小保持时间后才输出； n=1：DTMF信号的高电平保持时间超过配置项ReceiveDtmfSensitive 指定的最小保持时间后立即输出。 缺省值为 0。
功能描述	设置DTMF检测器输出DTMF字符的时机。有关此配置项的详细说明请参见第 1 章中DTMF脉宽滤波器部分。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡；

3.1.2.15 设置 DTMF 发生器(CTI 系列)

3.1.2.15.1 设置发送缓冲区的大小

配置项	**TxDtmfBufSize**
节	**[SystemConfig]**
书写格式	TxDtmfBufSize=n
取值范围	n≥1，缺省值为 50 个字符。
功能描述	设置 DTMF 发生器的发送缓冲区大小。

3.1.2.15.2 设置 DTMF 发生器的参数

配置项	**TxDtmfTimePara**
节	**[BoardId=x]**
书写格式	格式 1：TxDtmfTimePara={t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L} 格式 2：TxDtmfTimePara={t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L} 格式 3：TxDtmfTimePara={t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L}, {t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L},{t_H,t_L} 格式 4：TxDtmfTimePara={t_H,t_L}
取值范围	t_H≥40：高电平保持时间，单位为毫秒，必须为 16 毫秒的整数倍，缺省值为 80。 t_L ≥40：低电平保持时间，单位为毫秒，必须为 16 毫秒的整数倍，缺省值为 80。
功能描述	设置DTMF发生器产生的DTMF字符的高电平保持时间和低电平保持时间。有关此配置项的详细说明请参见第 1 章中DTMF发生器部分。
注意事项	● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列； ● 格式 4 适用于 SHD 系列。

配置项	DefaultTxDelayTime
节	[SystemConfig]
书写格式	DefaultTxDelayTime=n
取值范围	n：保持静默的时间，单位为毫秒，缺省值为 2000。
功能描述	设置发送DTMF时延时字符的延时时间。有关此参数的含义请参见SsmTxDtmf函数的说明。

3.1.2.16 设置回波抵消器(CTI 系列)

3.1.2.16.1 设置工作状态

配置项	EnableEchoCancellor
节	[BoardId=x]
书写格式	格式 1：EnableEchoCancellor=b,b,b,b 格式 2：EbableEchoCancellor=b,b,b,b,b,b,b,b 格式 3：EnableEchoCancellor=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：EnableEchoCancellor=b
取值范围	b=0：关闭； b=1：开启（缺省值）。
功能描述	设置回波抵消器的工作状态。详细内容请参见第 1 章中回波抵消器部分。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列； ● 格式 4 适用于 SHD 系列板卡；

3.1.2.16.2 设置工作参数

配置项	EchoCancelDelaySize
节	[BoardId=x]
书写格式	EchoCancelDelaySize=k
取值范围	k：滤波延迟系数，取值范围及缺省值取决于配置项LoadFskBin的设置： LoadFskBin=0：0≤k≤7， 缺省值：6； LoadFskBin>0：0≤k≤31，缺省值：12。
功能描述	设置回波抵消器的滤波延迟系数。
注意事项	只支持 SHD 系列板卡和 SHT 系列的 USB 语音盒。

3.1.2.17 设置 Barge-in 检测器 (CTI 系列)

3.1.2.17.1 设置 Barge in 检测器的启动方式

配置项	AlwaysDetectBargeIn
节	[BoardId=x]
书写格式	AlwaysDetectBargeIn=b
取值范围	b=0：由驱动程序自动控制。驱动程序的调度机制为：当通道状态迁移到 S_CALL_TALKING，或者以动态发言的身份加入会议室时，自动开启 BargeIn 检测器；当通道状态迁移出 S_CALL_TALKING，或者具有动态发言身份的成员退出会议室时，自动停止 BargeIn 检测器； b=1：总是开启。 缺省值为 0。
功能描述	设置Barge in 检测器的启动方式。
注意事项	● 本配置项只适用于 SHD 系列板卡。

3.1.2.17.2 设置 Barge in 检测器的工作参数

配置项	VoiceEnergyMinValue
节	[BoardId=x]
书写格式	VoiceEnergyMinValue=k

取值范围	k：噪声判定门限，缺省值：100,000。
功能描述	设置 Barge in 检测器的噪声判定门限。

配置项	**BargeInSensitive**
节	**[BoardId=x]**
书写格式	格式 1：BargeInSensitive=k,k,k,k 格式 2：BargeInSensitive=k,k,k,k,k,k,k,k 格式 3：BargeInSensitive=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k 格式 4：BargeInSensitive=k
取值范围	0≤k≤31，值越大越灵敏，缺省值：6。
功能描述	设置 Barge In 检测器的灵敏度。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡； ● 格式 4 适用于 SHD、DTP 系列板卡； ● 详细内容请参见函数SsmSetBargeInSens的说明。

配置项	**BargeInDtrmTime**
节	**[BoardId=x]**
书写格式	格式 1：BargeInDtrmTime=k,k,k,k 格式 2：BargeInDtrmTime=k,k,k,k,k,k,k,k 格式 3：BargeInDtrmTime=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k 格式 4：BargeInDtrmTime=k
取值范围	k≥16，必须是 16ms 的整数倍，单位：毫秒，缺省值：32。
功能描述	设置 Barge In 的最小保持时间。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡； ● 格式 4 适用于 SHD、DTP 系列板卡； ● 详细内容请参见函数SsmSetIsBargeInDtrmTime的说明

配置项	**IsNoSoundDtrmTime**
节	**[SystemConfig]**
书写格式	IsNoSoundDtrmTime=k
取值范围	最小保持时间，单位：毫秒，取值范围：≥16，必须为 16 毫秒的整数倍，缺省值为 5000 毫秒。
功能描述	设置静默的最小保持时间。更多信息请参见函数SsmSetNoSoundDtrmTime的说明。
使用范例	IsNoSoundDtrmTime= 5000

配置项	**DefaultDtmfIsSound**
节	**[BoardId=x]**
书写格式	格式 1：DefaultDtmfIsSound=b,b,b,b 格式 2：DefaultDtmfIsSound=b,b,b,b,b,b,b,b 格式 3：DefaultDtmfIsSound=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：DefaultDtmfIsSound=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否将来话信号中的 DTMF 信号视为语音信号。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡； ● 格式 4 适用于 SHD、DTP 系列板卡； ● 本配置项要求 SynCTI Ver. 4.5.2.9 及以后版本。

3.1.2.18 设置 TDM 总线连接

配置项	InVoiceToBus
节	[BoardId=x]
书写格式	格式 1：InVoiceToBus=b,b,b,b 格式 2：InVoiceToBus=b,b,b,b,b,b,b,b 格式 3：InVoiceToBus=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：InVoiceToBus=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	系统初始化时，是否将来话信号连接到TDM交换总线。函数SsmSetFlag（携带F_InVoiceToBus参数）可以实现相同功能。
注意事项	● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡； ● 格式 4 适用于 SHD、DTP 系列板卡； ● 本配置项要求 SynCTI Ver. 4.0 及以后版本。

配置项	LinkFromStopPlayAndTone
节	[BoardId=x]
书写格式	LinkFromStopPlayAndTone=b
取值范围	b=1：立即终止放音任务或信号音发生器任务（缺省值）； b=0：等待放音任务终止后，再执行下总线操作命令。
功能描述	详细内容请参见第 1 章中“SHD系列板卡的原理框图”部分内容。
注意事项	● 本配置项只适用于 30 通道、60 通道和 120 通道规格的 SHD 系列板卡； ● 本配置项要求 SynCTI Ver. 4.5.2.9 及以后版本。

配置项	DefaultSpeakVolume
节	[BoardId=x]
书写格式	格式 1：DefaultSpeakVolume=k,k,k,k 格式 2：DefaultSpeakVolume=k,k,k,k,k,k,k,k 格式 3：DefaultSpeakVolume=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k
取值范围	-6≤k≤6，乘以 3 即为分贝，大于 0 表示音量提升，小于 0 表示音量衰减，缺省值为 0。
功能描述	设置来话信号进入总线前的音量调节器 A5，详细内容请参见第 1 章中板卡的工作原理框图部分。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒、ATP 系列的 USB 录音盒； 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡、DST 系列板卡。

3.1.2.19 设置模拟电话线的 Caller ID 检测器（SHT＋ATP 系列）

3.1.2.19.1 设置 Caller ID 检测器的工作模式

配置项	CallerIdStyle
节	[BoardId=x]
书写格式	格式 1：CallerIdStyle=m,m 格式 2：CallerIdStyle=m,m,m,m 格式 3：CallerIdStyle=m,m,m,m,m,m,m,m 格式 4：CallerIdStyle=m,m,m,m,m,m,m,m,m,m,m,m,m,m,m,m
取值范围	m=0：DTMF制式，配置项DtmfCallerIDStyleLength和DtmfCallerIDInterTimeout有效； m=1：FSK制式，配置项CloseCallerIdOnReceived和FSKCallerIdDtrmTime有效。 缺省值为 1。
功能描述	设置对端交换机在模拟电话线上发送主叫号码的模式。详细信息请参见第 1 章中的“模拟电话线上的主叫号码”部分内容。

注意事项	● 本配置项只适用于 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的录音通道； ● 格式 1、格式 2 分别适用于 2 通道/4 通道规格 USB 语音盒和录音盒，格式 3 适用于 8 通道规格的板卡，格式 4 适用于 16 通道规格的板卡； ● 本配置项实现的功能也可以通过调用函数SsmSetFlag（携带F_CALLERIDSTYLE参数）实现。

3.1.2.19.2 设置 FSK 制式的工作参数

配置项	**CloseCallerIdOnReceived**
节	**[SystemConfig]**
书写格式	CloseCallerIdOnReceived=b
取值范围	b=0：不关闭； b=1：关闭（缺省值）。
功能描述	如果Caller ID检测器工作在FSK制式，当驱动程序完成FSK制式的主叫号码的接收后，本配置项设置是否由驱动程序自动关闭Caller ID检测器。 更多信息请参见第 1 章中“模拟电话线上的主叫号码”部分内容。
注意事项	本配置项只适用于 SHT、ATP 系列板卡。

配置项	**FSKCallerIdDtrmTime**
节	**[BoardId=x]**
书写格式	格式 1：FSKCallerIdDtrmTime=k,k 格式 2：FSKCallerIdDtrmTime=k,k,k,k 格式 3：FSKCallerIdDtrmTime=k,k,k,k,k,k,k,k 格式 4：FSKCallerIdDtrmTime=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k
取值范围	k：结束的判断时间，单位为毫秒，100≤k≤500，缺省值为 500ms。
功能描述	FSK 制式的 Caller ID 检测器启动后，由于 FSK 制式的 Caller ID 号码在传送时没有明确的结束标志，因此，驱动程序只能依靠时间与数据的相关性来判断传送过程的结束，即，当收到的 Caller ID 字符的长度在经过了本参数设定的时间后，仍然保持不变，则驱动程序认为 FSK 制式的 Caller ID 信号传送结束。
注意事项	● 本配置项只适用于 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的录音通道； ● 格式 1/格式 2 分别适用于 2 通道/4 通道规格 USB 语音盒和录音盒，格式 3 适用于 8 通道规格的板卡，格式 4 适用于 16 通道规格的板卡。

配置项	**FileterInvalidCID**
节	**[BoardId=x]**
书写格式	格式 1：FileterInvalidCID=b,b,b,b,b,b,b,b 格式 2：FileterInvalidCID=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否开启“过滤错误的 FSK 方式主叫号码”功能。驱动程序收到的 FSK 方式主叫号码可能会发生错误： ✧ 号码并非对端交换机发送，而是由于声音中的回波而形成的假号码； ✧ 号码的确是由对端交换机发送，但在传输过程中由于干扰等因素，信号被破坏，形成了不完整的号码。
注意事项	● 格式 1、格式 2 分别适用于 8 通道、16 通道规格的 SHT/ATP 系列板卡； ● 本配置项要求 SynCTI 驱动程序的版本为 Ver. 4.7.1.8 或更高。

配置项	**FskNoPhase**
节	**[BoardId=x]**
书写格式	FskNoPhase=m
取值范围	m=0：前导码为 0/1 翻转波形，然后是全 1 标志码（缺省值）； m=1：前导码为 0804，然后是全 1 标志码。
功能描述	设置 FSK 主叫号码信号中前导码的格式 。
注意事项	本配置项只适用于 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的录音通道。

3.1.2.19.3 设置 DTMF 制式的工作参数

配置项	**DtmfCallerIDStyleLength** **DtmfCallerIDInterTimeOut**
节	**[SystemConfig]**
书写格式	DtmfCallerIDStyleLength=n DtmfCallerIDInterTimeOut=t
取值范围	n：字符的最小个数，1≤n≤29，缺省值为 3。 t： 时间间隔，单位为毫秒，t≥300，缺省值为 500。
功能描述	当 Caller ID 检测器工作在 DTMF 制式时，如果驱动程序收到的主叫号码个数小于或等于 n，已经收到的主叫号码将被抛弃；每当驱动程序收到一个 DTMF 字符时，如果此字符与上一个字符之间的时间间隔大于 t，则以前收到的字符都将被抛弃。
注意事项	● DtmfCallerIDInterTimeOut 要求驱动程序的版本为 4.5.2.1 或更高； ● 本配置项只适用于 SHT、ATP 系列板卡。

3.1.2.20 设置铃流信号检测器（SHT+ATP 系列）

配置项	**RingDetectFilterPara**
节	**[SystemConfig]**
书写格式	RingDetectFilterPara=t_H, t_L
取值范围	t_H：高电平的最小保持时间，单位为 8 毫秒的倍数，t_H≥4，缺省值为 6，即 48 毫秒。只有当铃流信号保持高电平，并且保持时间大于或等于 t_H 时，驱动程序才会认为检测到了一次铃流信号。t_H 越大，铃流信号检测器的灵敏度越低。 t_L：低电平的最小保持时间，单位为 8 毫秒的倍数，t_L≥1，缺省值为 40，即 320 毫秒。只有当铃流信号保持低电平，并且保持时间大于或等于 t_L 时，驱动程序才会断定铃流信号消失。
功能描述	设置铃流信号检测器的滤波参数。模拟中继线通道和磁石通道检测振铃的参数。驱动程序每隔 8 毫秒对模拟中继线通道和磁石通道的振铃信号进行采样。当实际低电平计数值大于或等于低电平判定值（第 2 个参数）时认为一次振铃信号结束；当驱动程序检测到模拟中继线通道和磁石通道的振铃信号计数大于或等于高电平判定值（第 1 个参数）时判决为一次振铃。加大振铃信号高电平判定值会使振铃检测灵敏度降低，反之增加。
注意事项	本配置项只适用于 SHT 系列的模拟中继线通道和磁石通道，以及 ATP 系列的模拟电话线录音通道。

配置项	**AlwaysToRingingOnRingCntX**
节	**[SystemConfig]**
书写格式	AlwaysToRingingOnRingCntX=k
取值范围	k=0：驱动程序检测到 1 个（如果 Caller ID 检测器设置为 DTMF 模式）或 2 个（如果 Caller ID 检测器设置为 FSK 模式）完整的铃流信号后迁移到“振铃”状态； k≥1：无论 Caller ID 检测器工作于什么模式，驱动程序检测到 k 个完整的铃流信号后就迁移到“振铃”状态。 缺省值为 0。
功能描述	设置铃流信号检测器的工作参数。详细信息请参见第 1 章中“模拟电话线上的铃流信号”部分内容。
注意事项	● 本配置项可能会被配置项ChToRingingOnRingCnt所覆盖，详细信息请参见第 1 章中“模拟电话线上的铃流信号”部分内容； ● 本配置项设定的参数也可以通过函数SsmSetFlag（携带F_ChToRingingOnRingCnt参数）进行设置； ● 本配置项只支持 SHT 系列板卡的模拟中继线通道。

配置项	**ChToRingingOnRingCnt**
节	**[BoardId=x]**
书写格式	格式 1：ChToRingingOnRingCnt=n,n,n,n 格式 2：ChToRingingOnRingCnt=n,n,n,n,n,n,n,n 格式 3：ChToRingingOnRingCnt=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n

取值范围	n=0：驱动程序检测到 1 个（如果 Caller ID 检测器设置为 DTMF 模式）或 2 个（如果 Caller ID 检测器设置为 FSK 模式）完整的铃流信号后迁移到“振铃”状态； n≥1：无论 Caller ID 检测器工作于什么模式，驱动程序检测到 n 个完整的铃流信号后就迁移到“振铃”状态。 缺省值为 0。
功能描述	设置铃流信号检测器的工作参数。详细信息请参见第 1 章中“模拟电话线上的铃流信号”部分内容。
注意事项	● 本配置项只适用于 SHT 系列板卡的模拟中继线通道； ● 如果在某个通道上指定了本配置项，该通道基于配置项AlwaysToRingingOnRingCntX的参数将被覆盖，详细信息请参见第 1 章中“模拟电话线上的铃流信号”部分内容； ● 本配置项设定的参数也可以通过函数SsmSetFlag（携带F_ChToRingingOnRingCnt参数）进行设置； ● 格式 1 适用于 ATP 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 ATP 系列板卡。

3.1.2.21 设置 FSK 数据收发器(CTI 系列)

3.1.2.21.1 设置 FSK 数据发送器的工作参数

配置项	**FreqBit0** **FreqBit1** **Baudrate** **MdlAmp**
节	**[FskConfig]**
书写格式	FreqBit0=f_0 FreqBit1=f_1 Baudrate=k MdlAmp=a
取值范围	f_0：比特 0 的载波频率，单位为 Hz，300≤f_0≤3400，缺省值为 2200Hz。 f_1： 比特 1 的载波频率，单位为 Hz，300≤f_1≤3400，缺省值为 1200Hz。 k： 波特率，单位为 bps，300≤k≤2400，缺省值为 1200bps。 a： 信号的幅度，0≤n≤255，缺省值为 128。
功能描述	FreqBit0、FreqBit1 分别设置比特 0 和比特 1 的载波频率；Baudrate设置波特率；MdlAmp设置调制信号的幅度。有关FSK数据发送器的更多信息请参见第 1 章中“FSK数据收发器”部分内容。

3.1.2.22 设置 FSK 数据接收器(CTI 系列)

配置项	**FskMarkSignal**
节	**[BoardId=x]**
书写格式	格式 1：FskMarkSignal=m_0,m_1,m_2,m_3,m_4,m_5,m_6,m_7 格式 2：FskMarkSignal=m_0,m_1,m_2,m_3,m_4,m_5,m_6,m_7,m_8,m_9,m_{10},m_{11},m_{12},m_{13},m_{14},m_{15} 格式 3：FskMarkSignal=m
取值范围	m, m_i：标志码的个数（下标 i 表示通道的物理编号），可以使用的值为： =0：标志码为连续 12 个 1（缺省值）； =1：标志码为连续 8 个 1。
功能描述	设置 FSK 数据流中标志码（前导信号）的格式。
注意事项	格式 1、格式 2 分别适用于 8 通道、16 通道规格的 SHT 系列板卡； 格式 3 适用于 SHD 系列板卡。

配置项	**FskEchoCancelDelay**
节	**[BoardId=x]**
书写格式	格式 1：FskEchoCancelDelay=b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9,b_{10},b_{11},b_{12},b_{13},b_{14},b_{15} 格式 2：FskEchoCancelDelay=b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7

取值范围	b_i=0：关闭（缺省值）； b_i=1：开启。 注：下标 i 为通道的物理编号。
功能描述	当 FSK 接收器处于工作状态时，是否关闭回波抵消器。
注意事项	● 格式 1、格式 2 分别适用于 8 通道、16 通道规格的 SHT 系列板卡； ● 当通道上启动了FSK接收器后，回波抵消器的工作状态由本配置项决定，配置项EnableEchoCancellor被忽略；否则，回波抵消器的工作状态由配置项EnableEchoCancellor决定，本配置项被忽略； ● 本配置项要求 SynCTI 的版本为 Ver4.5.6.2 或更高； ● 函数SsmSetFlag（携带F_EchoCancelInFsk参数）可以实现相同功能。

3.1.2.23 设置传真资源通道的工作参数

配置项	**MaxSpeed**
节	**[Fax]**
书写格式	MaxSpeed=k
取值范围	k：可用的传真速率，单位为 bps，可以使用的值为 4800、9600、14400 缺省值为 9600。
功能描述	设置接收或发送传真时使用的最大速率。

配置项	**RcvDisTime**
节	**[Fax]**
书写格式	RcvDisTime=n
取值范围	0≤n≤10，缺省值为 4。
功能描述	传真过程开始后，如果出现检测不到 Dis 信号或 Dis 信号无效的情况，本配置项设置重复检测 Dis 信号的次数。该配置项在 Ver4.7.2.0 版本上支持。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 4.7.2.0 或更高

配置项	**ResendForRTN**
节	**[Fax]**
书写格式	ResendForRTN=k
取值范围	k：0 收到 RTN 消息后不重发页面数据 （默认） k：1 收到 RTN 消息后重发页面数据
功能描述	设置在传真发送过程中收到 RTN 消息时是否进行页面重发

配置项	**ResetRcvForRTN**
节	**[Fax]**
书写格式	ResetRcvForRTN =k
取值范围	k：0 传真接收到效果不好的页面时，发送 RTN 消息（缺省） k：1 传真接收到效果不好的页面时，不发送 RTN，直接挂断
功能描述	在传真接收过程中接收到效果不好的页面数据时，判断是发送 RTN 还是直接挂断

配置项	**KeepPageForRTN**
节	**[Fax]**
书写格式	KeepPageForRTN =k
取值范围	k：0 传真接收到效果不好的页面时，不保留此页面 k：1 传真接收到效果不好的页面时，保留此页面（默认）
功能描述	传真接收到效果不好的页面数据时，判断是否保留此页面

配置项	**PercentageForRTN**
节	**[Fax]**
书写格式	PercentageForRTN =n
取值范围	0≤n≤100，缺省值为 30。
功能描述	当接收到的页面数据大于此百分比时，认为页面效果不好，否则认为页面数据正常

3.1.2.24 设置电压检测器

3.1.2.24.1 忽略电压检测器的检测结果

配置项	**IgnoreLineVoltage**
节	**[BoardId=x]**
书写格式	格式 1：IgnoreLineVoltage=b,b,b,b 格式 2：IgnoreLineVoltage=b,b,b,b,b,b,b,b 格式 3：IgnoreLineVoltage=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否忽略线路上的电压检测结果。详细内容请参见第 1 章中的“模拟电话线的电压变化”部分。
注意事项	● 本配置项只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）； ● 格式 1 适用于 ATP 系列的 USB 语音盒/录音盒； ● 格式 2 适用于 8 通道规格的 ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 ATP 系列板卡。
相关函数	SsmSetIgnoreLineVoltage

3.1.2.24.2 设置摘机/挂机判决条件

配置项	**IsHangupDtrmVoltage**
节	**[BoardId=x]**
书写格式	格式 1：IsHangupDtrmVoltage=k,k,k,k 格式 2：IsHangupDtrmVoltage=k,k,k,k,k,k,k,k 格式 3：IsHangupDtrmVoltage=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k
取值范围	k：判定电压，单位为伏（V），缺省值为 26V。
功能描述	设置ATP系列板卡的录音通道的摘/挂机判定电压。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。
注意事项	● 本配置项只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）； ● 格式 1 适用于 ATP 系列的 USB 语音盒/录音盒； ● 格式 2 适用于 8 通道规格的 ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 ATP 系列板卡。
相关函数	SsmSetDtrmLineVoltage

配置项	**LineOncnt**
节	**[SystemConfig]**
书写格式	LineOncnt=t
取值范围	t：最小保持时间，t≥1，每个单元表示 8 毫秒，缺省值为 25（即 200 毫秒）。
功能描述	当线路电压符合摘机判定电压的条件，并且保持本配置项设定的最小保持时间后，驱动程序才认为线路上发生了摘机动作。更多信息请参见第 1 章中“模拟电话线的电压变化”部分内容。
注意事项	● 本配置项只适用于 ATP 系列板卡，或 SHT 系列板卡上的录音模块； ● 本配置项要求驱动程序的版本为 Ver. 4.7.2.0 或更高。

3.1.2.24.3 设置断线监测的判定条件

配置项	**OffLineSet**
节	**[SystemConfig]**
书写格式	OffLineSet=k
取值范围	k：电压门限值，单位为伏特（V），缺省值为 5。
功能描述	设置模拟电话线的断线判定电压门限值。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。
注意事项	本配置项只适用于模拟中继线的录音通道。

配置项	**OffLineDetermineTime**
节	**[SystemConfig]**
书写格式	OffLineDetermineTime=t

取值范围	t：最小保持时间，单位为毫秒，缺省值为 200。
功能描述	设置判定模拟电话线断开的最小保持时间。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。
注意事项	本配置项只适用于 ATP 系列板卡的录音通道和 SHT 系列板卡的模拟中继线通道。

3.1.2.24.4 设置极性反转信号是否影响通道的状态迁移

配置项	**DisablePolarReverse**
节	**[BoardId=x]**
书写格式	格式 1：DisablePolarReverse=b,b,b,b 格式 2：DisablePolarReverse=b,b,b,b,b,b,b,b 格式 3：DisablePolarReverse=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：DisablePolarReverse= [0,29]:b
取值范围	b=0：影响（缺省值）； b=1：不影响。
功能描述	设置极性反转信号是否影响通道的状态迁移。详细内容请参见第 1 章中“模拟电话线的电压变化”部分。
注意事项	● 本配置项只适用 SHT 系列板卡的模拟中继线通道； ● 格式 1 适用于 SHT 系列的 USB 语音盒/录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于 30 通道规格的远端坐席接入模块。

3.1.2.25 设置 USB 录音盒/语音盒中的内置板载扬声器

配置项	**USBLine0Output**
节	**[BoardId=x]**
书写格式	USBLine0Output=m
取值范围	m=0：将声音送到板载扬声器进行播放； m=1：将声音送到板载扬声器插座，由外置扬声器进行播放。
功能描述	开启或关闭USB录音盒/语音盒中的内置板载扬声器。更多信息请参见第 1 章中“ATP系列板卡的工作原理图”或“SHT系列板卡的工作原理图”部分内容。
注意事项	● 本配置项只适用于 ATP 系列的 USB 录音盒和 SHT 系列的 USB 语音盒； ● 本配置项要求 SynCTI 驱动程序的版本为 4.7.1.7 或更高。

3.1.2.26 SHT 系列板卡常用配置项(CTI 系列)

3.1.2.26.1 模拟中继通道专用配置项

3.1.2.26.1.1 设置去话呼叫的参数

配置项	**MaxWaitDialToneTime**
节	**[SystemConfig]**
书写格式	MaxWaitDialToneTime=t
取值范围	t：最大等待时间，单位为秒，0≤t≤30，缺省值为 3 秒。
功能描述	设置等待拨号音的最大等待时间。 应用程序在调用函数SsmAutoDial函数启动AutoDial任务后，驱动程序需要首先检测拨号音。如果在本配置项指定的时间内不能检测到拨号音，AutoDial任务失败。 有关AutoDial任务的执行过程请参见第 1 章中“模拟中继线通道的状态转移”部分内容。
注意事项	本配置项只适用于模拟中继线通道。

3.1.2.26.2 设置闪断信号及保持时间

配置项	**DefaultTxFlashChar**
节	**[SystemConfig]**
书写格式	DefaultTxFlashChar =s
取值范围	s：闪断表示符，缺省值为“！”，不能为标准 DTMF 字符或“？”。
功能描述	设置发送闪断的表示符。当对函数SsmTxDtmf 传入该字符后，即发送一个闪断信号。

配置项	**DefaultTxFlashTime**
节	**[SystemConfig]**
书写格式	DefaultTxFlashTime=t
取值范围	32≤t≤1000，单位为毫秒，缺省值为 500。
功能描述	设置使用'!'字符作为参数调用函数SsmTxDtmf在模拟中继线上产生闪断信号时，闪断信号的保持时间。详细内容请参见第 1 章中“在模拟线路上产生闪断信号”部分内容。

3.1.2.26.2.1 远端摘机检测器

3.1.2.26.2.1.1 设置普通的远端摘机检测器的工作参数

配置项	**WaitAfterDialTime**
节	**[SystemConfig]**
书写格式	WaitAfterDialTime=t
取值范围	t：最大等待时间，单位为毫秒，16≤t≤20000，缺省值为 3000。
功能描述	应用程序在模拟中继线通道上调用SsmAutoDial函数后，如果驱动程序向对端交换机发送完全部被叫号码后没有在线路上检测到回铃音，也没有在本配置项设定的时间内检测到任何语音活动，驱动程序会输出E_CHG_ToneAnalyze事件（携带CHKTONE_NOVOICE参数）。更多信息请参见第 1 章中“普通的远端摘机检测器”部分内容。
注意事项	本配置项只适用于 SHT 系列板卡的模拟中继线通道。

配置项	**MaxWaitVocAfterEcho**
节	**[SystemConfig]**
书写格式	MaxWaitVocAfterEcho=t
取值范围	t：最大等待时间，单位为秒，缺省值为 0。
功能描述	应用程序在模拟中继线通道上调用SsmAutoDial函数后，如果驱动程序向对端交换机发送完全部被叫号码后，在线路上检测到了回铃音，但没有在本配置项设定的时间内检测到任何语音活动，驱动程序会输出E_CHG_ToneAnalyze事件（携带CHKTONE_ECHO_NOVOICE参数）。更多信息请参见第 1 章中“普通的远端摘机检测器”部分内容。
注意事项	本配置项只适用于 SHT 系列板卡的模拟中继线通道。

配置项	**VoiceOnDetermineTime**
节	**[SystemConfig]**
书写格式	VoiceOnDetermineTime=t
取值范围	16≤t≤1024，单位为毫秒，必须为 16 的整数倍，缺省值为 96ms。
功能描述	设置语音活动信号的最小保持时间。只有当线路上连续出现语音活动信号，并且保持的时间大于本配置项的设定值时，驱动程序才会确认线路上出现了真正的语音活动信号。更多信息请参见第 1 章中“普通的远端摘机检测器”部分内容。
注意事项	本配置项只适用于 SHT 系列板卡的模拟中继线通道。

3.1.2.26.2.1.2 设置增强的远端摘机检测器

- **设置工作状态**

配置项	**RelativeEngyHookDetect**
节	**[SystemConfig]**
书写格式	RelativeEngyHookDetect=b
取值范围	b=0：关闭； b=1：开启（缺省值）。
功能描述	设置模拟中继线通道上“增强的远端摘机检测器”的工作状态。更多信息请参见第 1 章中“增强的远端摘机检测器”部分内容。
注意事项	● 本配置项只适用于 SHT 系列板卡的模拟中继线通道，并且要求 SynCTI 的版本为 3.5.2.4 或更高； ● 本配置项实现的功能也可以通过调用函数SsmSetFlag（携带F_RELATIVEENGYHOOKDETECT参数）实现。

- **设置工作参数**

配置项	**HookEngyConfigMulti** **HookValidEngyCnt**
节	**[SystemConfig]**
书写格式	HookEngyConfigMulti=k HookValidEngyCnt=t
取值范围	k：灵敏度，2≤k≤65535，值越大，灵敏度越低，即漏检测的概率增加，而误检测的概率减小。缺省值为 6。 t：最小保持时间，单位为毫秒，32≤t≤65535，必须为 16 的整数倍，缺省值为 48ms。值越大，灵敏度越低，即漏检测的概率增加，而误检测的概率减小。
功能描述	设置模拟中继线通道上的“增强的远端摘机检测器”的工作参数，HookEngyConfigMulti 设置摘机检测的灵敏度，HookValidEngyCnt 设置最小保持时间，以消除信号抖动对判断结果造成的干扰。只有当通道上检测到了瞬时的摘机动作，并且保持时间大于或等于 t，驱动程序才会判定线路上发生了真正的摘机动作。 更多信息请参见第 1 章中“增强的远端摘机检测器”部分内容。
注意事项	● 本配置项只适用于SHT系列板卡的模拟中继线通道，只有当配置项RelativeEngyHookDetect设置为 1 时有效； ● 本配置项要求 SynCTI 的版本为 3.5.2.4 或更高。

3.1.2.26.3坐席通道专用配置项

配置项	**AutoSendDialTone**
节	**[BoardId=x]**
书写格式	格式 1：AutoSendDialTone=b,b,b,b 格式 2：AutoSendDialTone=b,b,b,b,b,b,b,b 格式 3：AutoSendDialTone=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：AutoSendDialTone=[S_1,E_1]:b+…+ [S_m,E_m]:b
取值范围	b=1：开启； b=0：关闭(缺省值)。 S_1：起始通道编号，缺省值为 0。 E_1：结束通道编号，缺省值为板卡上的总通道数。 m：参数段的数量，最小为 1，最大不能超过板卡上的总通道数。
功能描述	设置当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的功能。

注意事项	● 本配置项只适用坐席通道，包括安装了坐席模块的 SHT 系列板卡和连接到远端坐席接入模块的 SHD 系列板卡； ● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于连接到远端坐席接入模块的SHD系列板卡，详细信息请参见第 1 章中“连接到远端坐席接入模块”部分内容。
使用范例	SHD-30A-CT/PCI/SS1 板卡连接到了远端坐席接入模块，希望开启前 15 个通道的本项功能，关闭后 15 个通道的本项功能，可以按照下列方式设置： AutoSendDialTone=[0,14]:1+[15,29]:0 如果全部 30 个通道都设置为开启，可以按照下列方式设置： AutoSendDialTone=[0, 29]:1

配置项	**StopSendDialToneOnDtmf**
节	**[BoardId=x]**
书写格式	格式 1：StopSendDialToneOnDtmf=b,b,b,b 格式 2：StopSendDialToneOnDtmf=b,b,b,b,b,b,b,b 格式 3：StopSendDialToneOnDtmf=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4：StopSendDialToneOnDtmf =[S_1,E_1]:b+…+ [S_m,E_m]:b
取值范围	b=1：开启； b=0：关闭（缺省值）。 S_1：起始通道编号，缺省值为 0。 E_1：结束通道编号，缺省值为板卡上的总通道数。 m：参数段的数量，最小为 1，最大不能超过板卡上的总通道数。
功能描述	当驱动程序正在向话机发送信号音的时候，如果检测到了 DTMF 字符，本配置项设置是否启用由驱动程序自动停止发送拨号音的功能。
注意事项	● 本配置项只适用坐席通道，包括安装了坐席模块的 SHT 系列板卡和连接到远端坐席接入模块的 SHD 系列板卡； ● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于连接到远端坐席接入模块的SHD系列板卡，详细信息请参见第 1 章中“连接到远端坐席接入模块”部分内容。配置范例请参考配置项AutoSendDialTone中的范例。

配置项	**MaxLocalFlashTime**
节	**[SystemConfig]**
书写格式	MaxLocalFlashTime=t
取值范围	32≤t≤2000，单位为毫秒，缺省值为 700
功能描述	设置判断闪断信号的最大保持时间
注意事项	本函数只适用于坐席通道

配置项	**UserOnHookFilterTime**
节	**[BoardID=x]**
书写格式	格式 1：UserOnHookFilterTime=t,t,t,t 格式 2：UserOnHookFilterTime=t,t,t,t,t,t,t,t 格式 3：UserOnHookFilterTime=t,t,t,t,t,t,t,t,t,t,t,t,t,t,t,t 格式 4：UserOnHookFilterTime =[S_1,E_1]:t+…+ [S_m,E_m]:t
取值范围	t：摘机信号的最小保持时间，单位为毫秒，必须为 16 毫秒的整数倍，值越小越敏感。缺省值为 64。 S_1：起始通道编号，缺省值为 0。 E_1：结束通道编号，缺省值为板卡上的总通道数。 m：参数段的数量，最小为 1，最大不能超过板卡上的总通道数。
功能描述	设置摘机信号的最小保持时间。

注意事项	● 本配置项只适用坐席通道，包括安装了坐席模块的 SHT 系列板卡和连接到远端坐席接入模块的 SHD 系列板卡； ● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于连接到远端坐席接入模块的SHD系列板卡，详细信息请参见第 1 章中“连接到远端坐席接入模块”部分内容。配置范例请参考配置项AutoSendDialTone中的范例。 ● 本配置项的缺省值已经能够适应绝大多数的应用场合，通常无须设置。

配置项	**UserChGenerateRingMode**
节	**[BoardId=x]**
书写格式	UserChGenerateRingMode=m
取值范围	m=0：1 秒高电平、4 秒低电平方式（缺省值）； m=1：持续高电平，直到应用程序发出停止命令。
功能描述	设置坐席通道或磁石话机通道产生振铃信号的方式。

配置项	**UserSendPolar**
节	**[BoardId=x]**
书写格式	格式 1：UserSendPolar=b,b,b,b 格式 2：UserSendPolar=b,b,b,b,b,b,b,b 格式 3：UserSendPolar=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置坐席通道是否支持产生极性反转信号的功能。详细信息请参见第 1 章中“在电话线上产生极性反转信号”部分内容。
注意事项	● 本配置项只适用 SHT 系列板卡的坐席通道； ● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.26.4传真通道专用配置项

配置项	**MaxFaxChannel**
节	**[BoardId=x]**
书写格式	MaxFaxChannel=n
取值范围	取值范围及缺省值与板卡的型号有关，详细内容请参见第 1 章中“设置传真资源通道的数量”部分内容。
功能描述	设置传真资源通道的总数。

配置项	**DSP3WORKMODE**
节	**[BoardId=x]**
书写格式	DSP3WORKMODE=n
取值范围	n=1：板卡型号为 SHT-16B-CT/PCI/FAX（缺省值）； n=2：该板卡配置为硬件 GSM 录音方式（目前仅适用于 SHT-16B-CT/PCI/FAX 板卡）； n=3：板卡型号为 SHT-16B-CT/cPCI/MP3。
功能描述	设置型号为SHT-16B-CT/PCI/FAX或SHT-16B-CT/cPCI/MP3 的板卡的确切类型。更多信息请参见第 1 章中“SHT系列板卡简介”部分内容。

3.1.2.26.5联合模块的专用配置项

配置项	**UnimoduleState**
节	**[BoardId=x]**
书写格式	格式 1：UnimoduleState=b,b,b,b 格式 2：UnimoduleState=b,b,b,b,b,b,b,b 格式 3：UnimoduleState=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b

取值范围	b=0：是（缺省值）； b=1：否。
功能描述	设置是否将联合模块上的模拟中继线通道直接连接到坐席通道。
注意事项	● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.26.6 无业务模块通道的专用配置项

配置项	**NoModuleChBusRec**
节	**[BoardId=x]**
书写格式	格式 1：NoModuleChBusRec=b,b,b,b 格式 2：NoModuleChBusRec=b,b,b,b,b,b,b,b 格式 3：NoModuleChBusRec=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：关闭（缺省值）； b=1：开启。
功能描述	设置是否将来自总线的语音信号，而不是来话信号，作为Barge in检测器的输入信号源。更多信息请参见第 1 章中“使用未安装业务模块的通道”部分内容。
注意事项	● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.26.7 磁石通道

配置项	**IsMagnetModule**
节	**[BoardId=x]**
书写格式	格式 1：IsMagnetModule=b,b,b,b 格式 2：IsMagnetModule=b,b,b,b,b,b,b,b 格式 3：IsMagnetModule=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：模拟中继线模块（缺省值）； b=1：磁石模块。
功能描述	设置SHT系列板卡上安装的是模拟中继线模块还是磁石模块。更多信息请参见第 1 章中“磁石电话机通道”部分内容。
注意事项	● 格式 1 适用于 SHT 系列的 USB 语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.27 SHD 系列板卡常用配置项(CTI 系列)

3.1.2.27.1 设置来话呼叫的收号规则

配置项	**DefaultRcvPhoNumLen** **DefaultRcvCallerID** **RcvPhoNumCfgLen**
节	**[TUP] / [ISUP] / [ISDN] / [SS1Config]**
书写格式	DefaultRcvPhoNumLen=nz DefaultRcvCallerID=m RcvPhoNumCfgLen=b

取值范围	n：收号模式， n=0： 使用入局字冠方式的收号规则。入局字冠方式的收号规则由配置项MaxPhoNumRule、Rule进行设置； 1≤n≤40：使用指定长度的收号规则，n 为被叫号码的长度，在来话呼叫时，如果本端收齐了 n 个被叫号码，就认为收号过程结束。此时，配置项 DefaultRcvCallerID 和 RcvPhoNumCfgLen 有效。 n 的缺省值为 0。 m：是否需要接收主叫方号码， m=0：不需要（缺省值）； m=1：需要。 b： 被叫号码的保存模式， b=0：保存实际接收到的全部被叫号码（缺省值）； b=1：只保存前面的部分号码，个数由 DefaultRcvPhoNumLen 指定，多余部分被丢弃。
功能描述	DefaultRcvPhoNumLen 设置来话呼叫的收号规则的工作模式； DefaultRcvCallerID 设置是否需要接收主叫方的电话号码； RcvPhoNumCfgLen 设置驱动程序将接收到的被叫号码保存到内部缓冲区时的操作策略。
注意事项	● 只有当配置项 DefaultRcvPhoNumLen 设置为定长方式时，配置项 DefaultRcvCallerID 才有效； ● RcvPhoNumCfgLen 只适用于 TUP 通道； ● 本配置项只适用于 TUP 通道、ISUP 通道、ISDN 通道、SS1 通道； ● SS1 通道使用**[SS1Config]** 配置节； ● ISUP 通道使用**[ISUP]** 配置节； ● TUP 通道使用**[TUP]** 配置节； ● ISDN通道使用**[ISDN]** 配置节。对于ISDN通道，如果驱动程序工作在汇接局方式，本配置项设置的收号规则将被忽略。更多信息请参见第 1 章中“端局方式与汇接局方式”部分内容。

配置项	**MaxPhoNumRule** **Rule**
节	**[TUP] / [ISUP] / [ISDN] / [SS1Config]**
书写格式	MaxPhoNumRule=N 格式 1：Rule[i]="a…x",n,f,y,z 格式 2：Rule[i]="a…x",n,f,y 格式 3：Rule[i]="a…x",n,f
取值范围	N： 收号规则的个数。如果配置项的DefaultRcvPhoNumLen设定值大于 0，可以不设置本配置项；否则，N必须大于 0，并且配置项Rule必须从i=0 开始，重复设置N次。N的缺省值为 0。 i： 收号规则的编号，从 0 开始编号，0 ≤ i < N。 "a…x"：入局字冠的字符串，如“114”、“127”等。 n： 包括入局字冠的电话号码的总长度，只有当 y 设置为 0 时，本参数才有效。 f： 是否需要接收主叫方的电话号码， =1：需要； =0：不需要。 y： 电话号码的长度是否为固定值， =1：不固定，参数n和z无效。驱动程序完成局号的接收后，不再控制收号过程，将控制权交给应用程序。当应用程序完成全部号码的接收后，可以通过调用函数SsmSetKB来拒绝或接受来话呼叫，并将控制权交还给应用程序； =0：固定（缺省值），参数 n 和 z 有效。 z： 被叫号码的保存模式， z=0：保存实际接收到的全部被叫号码（缺省值）； z=1：只保存前面的部分号码，个数由 n 指定，多余部分被丢弃。
功能描述	MaxPhoNumRule 设置收号规则的个数； Rule 设置具体的收号规则，格式 1 只适用于 TUP 通道，格式 2 只适用于 TUP 通道和 ISUP 通道，格式 3 适用于 SS1 通道和 ISDN 通道

注意事项	● 只有当配置项DefaultRcvPhoNumLen设置为入局字冠方式时，本配置项才有效； ● 本配置项只适用于 TUP 通道、ISUP 通道、ISDN 通道、SS1 通道； ● SS1 通道使用**[SS1Config]** 配置节； ● ISUP 通道使用**[ISUP]** 配置节； ● TUP 通道使用**[TUP]** 配置节； ● ISDN通道使用**[ISDN]** 配置节。对于ISDN通道，如果驱动程序工作在汇接局方式，本配置项设置的收号规则将被忽略。更多信息请参见第 1 章中“端局方式与汇接局方式”部分内容。

配置项	**MfcLenCtrlPos** **MfcLengthTable** **CallerIdEnTable**
节	**[SS1Config]**
书写格式	MfcLenCtrlPos=n MfcLengthTable=m,m,m,m,m,m,m,m,m,m CallerIdEnTable=b,b,b,b,b,b,b,b,b,b
取值范围	n：被叫号码长度控制位索引，n≥1； m：被叫号码收号长度； b：是否需要接收主叫方的电话号码， =1：需要； =0：不需要。
功能描述	MfcLenCtrlPos 设置入中继通道被叫号码收号长度控制位； MfcLengthTable 设置入中继通道的 MFC 收号长度控制表； CallerIdEnTable 设置入中继通道接收主叫号码的使能控制表；以接收到的被叫号码的第 MfcLenCtrlPos 位作为索引，在收号长度控制表 MfcLengthTable 中查找相应的收号长度；同样，以此作为索引在主叫号码使能控制表 CallerIdEnTable 中可以查找到该号码是否需要接收主叫号码。
注意事项	● 本配置项只适用于使用 SS1 协议的通道。

3.1.2.27.2 设置“自动应答来话呼叫”功能

配置项	**AutoSendKB** **AutoSendACM** **UserSideAutoSendAck** **NetSideAutoSendAck**
节	**[SS1Config] / [TUP] / [ISUP] / [ISDN]**

书写格式	AutoSendKB=b // 使用[SS1Config]节
	AutoSendACM=b // 使用[TUP] 或 [ISUP]节
	UserSideAutoSendAck=b // 使用[ISDN]节
	NetSideAutoSendAck=b // 使用[ISDN]节

取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否开启“自动应答来话呼叫”功能。 AutoSendKB用于设置SS1 通道，更多信息请参见第 1 章中“中国 1 号信令的状态转移”部分内容； AutoSendACM用于设置ISUP通道和TUP通道，更多信息请参见第 1 章中“ISUP通道的状态转移”或“TUP 通道的状态转移”部分内容；UserSideAutoSendAck 用于设置 ISDN 通道（用户侧），NetSideAutoSendAck用于设置ISDN通道（网络侧），更多信息请参见第 1 章中“ISDN通道的状态转移图”中的k2 开关。
注意事项	● 函数SsmEnableAutoSendKB可以实现相同功能。

3.1.2.27.3设置去话呼叫的主叫号码

配置项	**TxCallerId** **CalloutCallerId**
节	**[SS1Config] / [TUP] / [ISUP] / [ISDN]**
书写格式	TxCallerId=s CalloutCallerId=s
取值范围	s：存放主叫方电话号码的 ASCII 码形式的字符串，ISDN 通道的最大字符数为 20，其它通道的最大字符数为 50，只能由数字字符'0'~'9'组成，其它类型的字符将被忽略。s 为空表示去话呼叫时不向对端交换机发送主叫号码。 缺省值为空。
功能描述	去话呼叫时，设置本端的电话号码，即主叫方电话号码。 TxCallerId 用于设置 SS1 通道(使用[SS1Config]配置节)；CalloutCallerId 用于设置 ISUP 通道（使用[ISUP]配置节）、ISUP 通道（使用[TUP]配置节）或 ISUP 通道（使用[ISDN]配置节）。
注意事项	● 对于TUP协议和ISUP协议，某些交换机要求主叫方发送的主叫号码字符串中应该包含ST信号（脉冲终了信号）。配置项SetSTSignal可以用来设置此项功能。

配置项	**SetSTSignal**
节	**[TUP] / [ISUP]**
书写格式	SetSTSignal=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	去话呼叫时，设置本端向对端交换机发送的主叫号码字符串中是否包含 ST 信号（脉冲终了信号）。
注意事项	● 本配置项只适用于使用 TUP 协议和 ISUP 协议的通道，ISUP 通道使用 [ISUP]配置节，TUP 通道使用[TUP] 配置节。

3.1.2.27.4连接到远端坐席接入模块

配置项	**UsageMode**
节	**[BoardId=x]**
书写格式	UsageMode=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否将SHD系列板卡连接到远端坐席接入模块(Channel bank)。详细信息请参见第 1 章中"连接到远端坐席接入模块"部分内容。
注意事项	本配置项只适用于 SHD 系列的下列型号板卡： ✓ SHD-30A-CT/PCI/SS1 ✓ SHD-30A-CT/PCI/ISDN ✓ SHD-30A-CT/PCI/SS7 ✓ SHD-30A-CT/cPCI ✓ SHD-30B-CT/PCI/FAX ✓ SHD-60A-CT/PCI/SS1 ✓ SHD-60A-CT/PCI/ISDN ✓ SHD-60A-CT/PCI/SS7 ✓ SHD-60A-CT/cPCI ✓ SHD-60B-CT/PCI/SS7/FAX ✓ SHD-60B-CT/cPCI/FAX ✓ SHD-120A-CT/PCI/SS1 ✓ SHD-120A-CT/PCI/ISDN ✓ SHD-120A-CT/PCI/SS7 ✓ SHD-120A-CT/cPCI

配置项	**CBProtocolType**
节	**[BoardId=x]**
书写格式	CBProtocolType=m

取值范围	m=0：CCS 方式； m=1：CAS 方式（缺省值）。
功能描述	设置将 SHD 系列板卡连接到远端坐席接入模块时使用的信令方式。
注意事项	● 本配置项只有在配置项UsageMode设置为 1 时有效。

配置项	**CBChannelType**
节	**[BoardId=x]**
书写格式	CBChannelType=m
取值范围	m=0：不使用此通道； m=1：坐席通道（缺省值）； m=2：E/M 通道。
功能描述	设置将 SHD 系列板卡连接到远端坐席接入模块时的通道类型。
注意事项	● 本配置项只有在配置项UsageMode设置为 1 时有效。

3.1.2.27.5设置板卡工作模式

配置项	**RunAsSpy**
节	**[BoardId=x]**
书写格式	RunAsSpy =b
取值范围	b=0：正常模式（缺省值）； b=1：监控模式。
功能描述	设置 SHD 系列板卡的工作模式。
注意事项	● 对于信令监控卡，始终处于监控模式，本配置对它无效。

3.1.2.27.6SS1 信令的高级设置

3.1.2.27.6.1 选择国家

配置项	**mfcr2_Protocol**
节	**[SS1Config]**
书写格式	mfcr2_Protocol=s

取值范围	s 为字符串，其含义如下表所示： s 的取值 / 支持的国家/协议 al 阿尔及利亚 ALGERIA ar 阿根廷 ARGENTINA bh 巴林 BAHRAIN bo 玻利维亚 BOLIVIA br 巴西 BRAZIL cl 智利 CHILE cn 中国 CHINA co-land 哥伦比亚陆地 COLOMBIA_LAND co-cell 哥伦比亚蜂窝 COLOMBIA_CELL cz 捷克 CZECH cd 刚果民主 DEMO_CONGO eg 埃及 EGYPT gn 加纳 GHANA hd 洪都拉斯 HONDURAS in 印度 INDIA id 印度尼西亚 INDONESIA iq 伊拉克 IRAQ itu ITU kr 朝鲜 KOREA kw 科威特 KUWAIT my 马来西亚 MALAYSIA mx 墨西哥 MEXICO ng 尼日利亚 NIGERIA pa 巴拿马 PANAMA ph 菲律宾 PHILIPINNES ro 罗马里亚 ROMANIA sa 沙特阿拉伯 SAUDI_ARABIA sg 新加坡 SINGAPORE th 泰国 THAILAND ve 委内瑞拉 VENEZUELA s 的缺省值为 cn。
功能描述	设置使用 No.1 信令的国家。
注意事项	只有当配置项ProtocolType设置为 0 时，本配置项才有效。

3.1.2.27.6.2 设置接续过程的基本工作参数

配置项	**TxCas_CD**
节	**[SS1Config]**
书写格式	TxCas_CD=k
取值范围	k：高 6 位比特保留，应设置为 0；低 2 比特为 C/D 位码： Bit1：C 的信令码，缺省值为 1； Bit0：D 的信令码，缺省值为 1。
功能描述	设置本端向对端交换机发送 ABCD 信令码时，CD 信令码的值。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

配置项	**RxCASFilterTime**
节	**[SS1Config]**
书写格式	RxCASFilterTime=t
取值范围	t：对端交换机送出的 ABCD 信令的最小保持时间，单位为毫秒，必须为 8 的整数倍，缺省值为 0。

功能描述	设置对端交换机送出的 ABCD 信令的最小保持时间。只有当线路上 ABCD 信令码发生变化，并且新值的保持时间超过本配置项的设定值，驱动程序才会确认 ABCED 信令码的变化，否则认为是干扰抖动信号。
注意事项	● 本配置项只适用于 SHD 系列板卡的 SS1 通道； ● 本参数适用于信号质量较差的 E1/T1 线路，当信令抖动非常严重时，往往会导致驱动程序误判远端挂机，本配置项的功能是设定一个最小保持时间以消除抖动。注意：对于去话呼叫，本配置项的设定值如果过大，可能会有副作用，即当被叫发送“被叫应答”信令（即 A_b/B_b=0/1）后，又立即发送“被叫先挂机”信令（即 A_b/B_b=1/1），则“被叫应答”信令有可能被滤波器当成抖动信号而过滤掉。

配置项	**MaxWaitMfcTime**
节	**[SS1Config]**
书写格式	MaxWaitMfcTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为 10 秒。
功能描述	设置 SS1 状态机的 T2 定时器。 有关T2 的更多信息请参见第 1 章中“中国 1 号信令的状态转移”部分内容。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

配置项	**RxR2FilterTime**
节	**[BoardId=x]**
书写格式	RxR2FilterTime=t
取值范围	t：R2 信号的最小保持时间，单位为毫秒，取值范围：16~96，必须为 16 的整数倍，缺省值为 16。
功能描述	设置 MFC R2 信号的最小保持时间。
注意事项	● 本配置项只适用于 SHD 系列板卡的 SS1 通道； ● 本配置项实现的功能也可以通过调用函数SsmSetFlag（携带F_RXR2FILTERTIME参数）实现。

3.1.2.27.6.3 设置状态机中通道的工作方式

配置项	**EnableAutoCall**
节	**[BoardId=x]**
书写格式	EnableAutoCall[n]=b
取值范围	n：数字中继线的板卡内部编号，0 ≤ n ≤ M-1，(M 为板卡的数字中继线总数)； b：是否使用 SynCTI 驱动程序提供的状态机， b=1：是（缺省值）； b=0：否。
功能描述	设置 SS1 通道状态机的工作状态。
注意事项	本配置项只适用于 SHD 系列的 SS1 通道。

配置项	**AutoCallInTimeSlot**
节	**[BoardId=x]**
书写格式	AutoCallInTimeSlot[n]=i,k
取值范围	n：数字中继线的板卡内部编号，0 ≤ n ≤ M-1，(M 为板卡的数字中继线总数)。 i：设置为来话方向的起始时隙。 k：设置为来话方向的通道（时隙）数量。
功能描述	将第 n 条 PCM 上从 i 时隙开始的 k 个时隙设置为来话方向，其余时隙设置为去话方向。
注意事项	本配置项只适用于 SHD 系列的 SS1 通道。

3.1.2.27.6.4 设置中国 1 号信令状态机的参数

配置项	**MfcKB**
节	**[SS1Config]**

书写格式	MfcKB=k
取值范围	1≤k≤6 ，缺省值为 1，表示被叫用户空闲，接受来话呼叫。有关KB取值的物理含义请参见函数SsmSetKB的说明。
功能描述	设置当 SS1 通道自动应答来话呼叫时向对端交换机发送的 KB 信号值。 详细内容请参见第 1 章中“中国 1 号信令的状态转移”部分内容。
注意事项	本配置项只有当配置项AutoSendKB设置为 1 时才有效。

配置项	**MaxWaitSetKBTime**
节	**[SS1Config]**
书写格式	MaxWaitSetKBTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为 3 秒。
功能描述	设置等候应用程序设置 KB 信号的最大时间。 设置SS1 通道状态机中应用程序应答本次来话呼叫请求的最大等待时间，即T4 定时器。有关T4 的更多信息请参见第 1 章中“中国 1 号信令的状态转移”部分内容。

配置项	**MaxWaitKDTime**
节	**[SS1Config]**
书写格式	MaxWaitKDTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为 60 秒。
功能描述	设置SS1 通道状态机中等待对端交换机发送KD信号的最大等待时间，即T3 定时器。有关T3 的更多信息请参见第 1 章中“中国 1 号信令的状态转移”部分内容。

配置项	**PhoNumHoldup** **A1ToA3pWaitTime** **A3pTime**
节	**[SS1Config]**
书写格式	PhoNumHoldup=m A1ToA3pWaitTime=t_{wait} A3pTime=t_{keep}
取值范围	m：“被叫号码拦截”功能的控制开关，其取值及含义如下表所示： **m** / **描述** 0：关闭“被叫号码拦截”功能 1：开启“被叫号码拦截”功能，只拦截 1 位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，不是向对端交换机发送互控方式的 A3 信号，而是发送 A1 信号，继续等待后续被叫号码。 ✧ 如果在 t_{wait} 时间内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，接收被叫号码的操作完成，驱动程序恢复向对端交换机发送互控方式的 A3 信号，并恢复正常的接续过程； ✧ 如果超过 t_{wait} 时间后仍未收到后续被叫号码，驱动程序向对端交换机发送脉冲方式的 A3 信号，并恢复正常的接续过程。 2：开启“被叫号码拦截”功能，拦截多位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，继续发送 A1 信号，等待后续被叫号码。如果在 t_{wait} 内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，并重复上述过程。直到超过 t_{wait} 后仍未收到下一个被叫号码，驱动程序才向对端交换机发送脉冲方式的 A3 信号，恢复正常的接续过程。 m 的缺省值为 0。 t_{wait}：最大等待时间，单位为毫秒，缺省值为 1000，m>0 时有效； t_{keep}：脉冲宽度，单位为毫秒，缺省值为 150，m>0 时有效。
功能描述	设置“被叫号码拦截”功能。PhoNumHoldup 设置“被叫号码拦截”功能的控制开关，A1ToA3pWaitTime 设置等待后续被叫号码的时间门限，A3pTime 设置脉冲形式的 A3 信号的脉冲宽度。
注意事项	PhoNumHoldup的功能也可以通过调用函数SsmSetFlag（携带F_RCVPHONUMHOLDUP参数）实现。

3.1.2.27.6.4.1 设置状态机中去话呼叫的参数

配置项	**MaxWaitOccupyAckTime**

节	**[SS1Config]**
书写格式	MaxWaitOccupyAckTime=t
取值范围	t≥1，单位为秒，缺省值为 60。
功能描述	设置 T5 定时器的值。 有关T5 定时器的详细信息请参见第 1 章中“中国 1 号信令的状态迁移”部分内容。

配置项	**MfcKD**
节	**[SS1Config]**
书写格式	MfcKD=k
取值范围	1≤k≤6，缺省值为 3（市内电话）。有关KD信号的详细说明请参见函数SsmSetKD的说明。
功能描述	设置去话呼叫时的发端业务类别，即 KD 信号。

配置项	**MfcKA**
节	**[SS1Config]**
书写格式	MfcKA=k
取值范围	1≤k≤10，缺省值为 1（普通/定期）。有关KA信号的详细说明请参见函数SsmSetKA的说明。
功能描述	设置去话呼叫时 KA 信号（本端的主叫类别）。

配置项	**MaxWaitKBTime**
节	**[SS1Config]**
书写格式	MaxWaitKBTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为 60 秒。
功能描述	设置等待对端交换机发送 KB 信号的最大等待时间。

3.1.2.27.6.4.2 与 Dialogic 的 SS1 通道对接

配置项	**ToRingingDelayTime**
节	**[BoardId=x]**
书写格式	ToRingingDelayTime=t
取值范围	t：延迟时间，单位为毫秒，缺省值为 0。
功能描述	设置延迟时间。在 Synway 板卡与 Dialogic 板卡的 SS1 通道对联时，如果 Synway 板卡的 SS1 通道作为呼入端，当完成全部被叫号码的接收后，需要在通道进入振铃状态之前插入一段等待时间。
注意事项	本配置项仅在将 Synway 的 SHD 系列板卡与 Dialogic 的板卡互联时使用。

配置项	**RepeatPhoNumOn1stR2bwdIsA5**
节	**[BoardId=x]**
书写格式	RepeatPhoNumOn1stR2bwdIsA5=m
取值范围	m=0：驱动程序设置挂起原因为 SS1OUT_BWD_A5，将通道迁移到 S_CALL_PENDING 状态； m=1：如果本端只发送了第 1 位被叫号码就收到对端的 A5 信号，则重新发送第 1 位号码；否则，驱动程序设置挂起原因为 SS1OUT_BWD_A5，将通道迁移到 S_CALL_PENDING 状态。 缺省值为 1。
功能描述	去话呼叫时，如果收到对端的 A5 信号（空号），本配置项选择驱动程序的后续动作。
注意事项	本配置项仅在将 Synway 的 SHD 系列板卡与 Dialogic 的板卡互联时使用。

3.1.2.27.6.4.3 应用程序退出时是否向闭塞对端

配置项	**IsBlockSS1In**
节	**[SS1Config]**
书写格式	IsBlockSS1In=b
取值范围	b=1：发送（缺省值）； b=0：不发送。
功能描述	当应用程序退出时，本配置项用于设置是否由驱动程序自动向对端交换机发送闭塞命令，以通知对端交换机不要再向本端发起呼叫。

注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.6.4.4 输出调试信息

配置项	**Ss1OutputLog**
节	**[SS1Config]**
书写格式	Ss1OutputLog=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否将接收和发送的 ABCD 位码输出到日志文件 Ss1Output.log 中。

配置项	**MfcR2ToRxCallerIdBuf**
节	**[SS1Config]**
书写格式	MfcR2ToRxCallerIdBuf=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否将呼叫过程中对端交换机发送的 R2 信号保存到 Caller ID 扩展缓冲区中，以便于对呼叫过程进行观察。如果开启此功能，每当对端交换机发送的 R2 信号发生变化，包括 R2 信号出现、R2 信号消失，驱动程序都会将此事件放入 RxCallerIdExBuf 缓冲区中，同时向应用程序抛出 E_CHG_CIDExBuf 事件。 函数SsmGetCallerIdEx用于取出该缓冲区中的字符串。

3.1.2.27.6.5 设置 LineSide 协议的参数

配置项	**LSWaitPickup**
节	**[SS1Config]**
书写格式	LSWaitPickup=t
取值范围	16≤t≤10000，单位为毫秒，必须为 8 的整数倍，缺省值为 96。如果设置的值超过取值范围，驱动程序自动将其修改为 96 毫秒。
功能描述	设置LineSide协议的状态机中的T1 定时器。有关T1 定时器的详细信息请参见第 1 章中 “Line side 协议的状态机描述”部分内容。
注意事项	本配置项只适用于使用Line side协议的SS1 通道，仅当配置项ProtocolType设置为 1 时有效。

配置项	**Ss1SendIdleState**
节	**[SS1Config]**
书写格式	Ss1SendIdleState=m
取值范围	m=1：使用 CAS01xx；（阿尔卡特交换机使用本值） m=0：使用 CAS00xx。
功能描述	设置向对端交换机发送空闲指示标志时使用的 ABCD 位码值。
注意事项	本配置项只适用于使用Line side协议的SS1 通道，仅当配置项ProtocolType设置为 1 时有效。

3.1.2.27.7 SS7 信令的高级设置

3.1.2.27.7.1 设置数字中继线第 16 时隙的属性

配置项	**UseTS16AsCircuit** **Ss7SignalingTS**
节	**[BoardId=x]**
书写格式	UseTS16AsCircuit[n]=b Ss7SignalingTS[n]=k

取值范围	n：数字中继线的物理编号，取值范围为 0~M-1，M为配置项PcmNumber的设定值。 b：本数字中继线内是否包含 SS7 信令链路， b=0：是(缺省值)。具体提供信令链路的时隙编号由配置项 Ss7SignalingTS 决定； b=1：否。配置项 Ss7SignalingTS 所指定的时隙既不能用于信令，也不能用于话路。 k：时隙编号，缺省值为 16。 1) 如果 SynCTI 驱动程序的版本低于 4.5.8.0，k 固定为 16； 2) 如果驱动版本高于或等于 4.5.8.0 且低于 4.8.0.0，k 的取值与板卡型号有关 若板卡型号为 SHD 系列中的下列型号之一: ✧ SHD-240A-CT/cPCI/SS7 ✧ SHD-240S-CT/cPCI/SS7 ✧ SHD-480A-CT/cPCI/SS7 ✧ SHD-480S-CT/cPCI/SS7 k 只能设置为 16；如果是其余型号，k 可以选择 1 或 16。 3) 如果驱动版本高于或等于 4.8.0.0 若板卡型号为 SHD 系列中的下列型号之一: ✧ SHD-240A-CT/cPCI/SS7 ✧ SHD-240S-CT/cPCI/SS7 ✧ SHD-480A-CT/cPCI/SS7 ✧ SHD-480S-CT/cPCI/SS7 k 只能设置为大于等于 0，且小于等于 32；取值等于 0 或 32 时本数字中继线 1-31 时隙全部用于话路，其他值为信令链路时隙编号； 若板卡型号为 SHD 系列中的下列型号之一: ✧ SHD-240C-CT/cPCI ✧ SHD-240C-CT/cPCI/EC ✧ SHD-480C-CT/cPCI ✧ SHD-480C-CT/cPCI/EC ✧ SHD-240D-CT/PCI ✧ SHD-240D-CT/PCI/EC ✧ SHD-120D-CT/PCI ✧ SHD-120D-CT/PCI/EC k 只能设置为大于 0，且小于 32； 其它 SHD 系列板卡 k 可以选择 1 或 16。
功能描述	UseTS16AsCircuit 设置本数字中继线内是否包含 SS7 信令链路。
注意事项	● 本配置项为高级配置项，只适用于 SS7 信令； ● 如果配置项PcmNumber大于 1，应为每条数字中继线对本配置项进行设置。

3.1.2.27.7.2 设置地址信号中备用值的对应字符

配置项	**AddressSignal**
节	**[TUP] / [ISUP]**
书写格式	AddressSignal[n]=c
取值范围	n：TUP/ISUP 消息中地址信号使用的值，10≤n≤14。 c：n 对应的编码字符，不能为'0'~'9'中的任何一个字符。
功能描述	设置地址信号中备用值的对应字符。 更多信息请参见第 1 章中"设置地址信号中备用值的对应字符"部分内容。
注意事项	● 本配置项只适用于 TUP/ISUP 通道； ● ISUP 通道使用**[ISUP]**配置节，TUP 通道使用**[TUP]**配置节。

3.1.2.27.7.3 设置 SS7 信令服务器的 IP 地址

配置项	**Ss7ServerIP** **SecondServerIP**
节	**[SS7]**
书写格式	Ss7ServerIP=a.b.c.d SecondServerIP=a.b.c.d
取值范围	a.b.c.d：SS7 信令服务器的 IP 地址，缺省值为 127.0.0.1（表示 SS7 信令服务器只有一个，并且与应用程序一起，都运行于本机上）。

功能描述	设置 SS7 信令服务器 IP 地址。
注意事项	如果 SS7 信令服务器程序与应用程序都在本机上运行，则无需设置这 2 个配置项。如果系统中包含了 2 个 SS7 信令服务器，则 Ss7ServerIP、SecondServerIP 分别用于设置主信令服务器、从信令服务器的 IP 地址。
使用范例	Ss7ServerIP=201.123.123.1

3.1.2.27.7.4 设置本机的 IP 地址

配置项	**LocalIP**
节	**[SS7]**
书写格式	LocalIP=a.b.c.d
取值范围	a.b.c.d：本机的 IP 地址，缺省值为 127.0.0.1，表示应用程序与 SS7 信令服务器都在本机运行。
功能描述	设置 SS7 信令服务器或网关的 IP 地址。
使用范例	LocalIP=201.123.123.5

3.1.2.27.7.5 SS7 信令服务器输出语音信道

配置项	**LoadShp_a3AsSIU**
节	**[SystemConfig]**
书写格式	LoadShp_a3AsSIU=b
取值范围	b=0：否(缺省值)； b=1：是。
功能描述	设置SS7 信令服务器是否输出语音时隙。当SS7 信令服务器上不运行任何业务系统，只是提供SS7 信令服务时，它只处理数字中继线上的信令时隙，不处理语音时隙。如果接入到SS7 信令服务器的数字中继线上除了信令链路外，还包含语音时隙，需要SS7 信令服务器提供将语音时隙交换输出的功能。输出语音时隙的原理为：将物理编号为 0 的数字中继线上的全部语音时隙分别与物理编号为 1 的数字中继线上的对应语音时隙进行双向连接。更多信息请参见第 4 章中“为第三方板卡提供SS7 信令服务”部分内容。
注意事项	只适用于 SHD-60A-CT/PCI/SS7 型号的板卡。

3.1.2.27.7.6 设置数字中继线 SS7 电路

配置项	Ss7CircuitMap[pcm]
节	**[BoardId=x]**
书写格式	Ss7CircuitMap[pcm]=b
取值范围	pcm：数字中继线在板卡上的物理编号，更多信息请参见相关手册的说明。取值范围：0≤pcm≤31；b：DWORD 32bit，从 bit0-bit31 分别指示数字中继 PCM 的 TS0-TS31 是否用作 SS7 电路，0 不用，1 使用。在 E1 系统中 TS0 及处理 SS7 信令的时隙将被忽略。本配置项缺省值 0xFFFFFFFF。
功能描述	设置数字中继线 PCM 的 SS7 电路与 PCM 时隙的对应关系。对于设置为不用作 SS7 电路的时隙，驱动将不对其进行复原，并且阻断对端有关该时隙的呼入，该时隙对应的通道状态为”不可用”。
使用范例	Ss7CircuitMap[0]= 0xFFFFFFFF

3.1.2.27.7.7 TUP 协议高级配置项

3.1.2.27.7.7.1 设置电路群消息的电路范围

配置项	**SendGRMRange**
节	**[TUP]**
书写格式	SendGRMRange=m

取值范围	m=0：使用全 0 范围编码字段。电路群消息的 CIC 字段的时隙为 1，范围字段为 0，表示该电路群消息覆盖了时隙 1~31； m=1：使用非全 0 范围编码字段。电路群消息覆盖的时隙范围与配置项Ss7SignalingTS有关。假定Ss7SignalingTS的设定值为N，则： ✧ 如果 N=1，电路群消息的 CIC 字段的时隙为 2，范围字段为 29； ✧ 如果 N=16，电路群消息会被拆分成 2 条，其中一条消息的 CIC 字段的时隙为 1，范围字段为 14，另外一条消息的 CIC 字段的时隙为 17，范围字段为 14。
功能描述	当本端驱动程序向对端交换机发送电路群消息时，本配置项设置电路群消息所覆盖的时隙范围。
注意事项	本配置项会影响所有的电路群消息，包括电路群复原消息、电路群闭塞/解除闭塞。

配置项	**HangupRingSendCBK**
节	**[TUP]**
书写格式	HangupRingSendCBK=m
取值范围	m=0：发送 CFL 消息（缺省值）； m=1：发送 CBK 消息。
功能描述	当通道处于S_CALL_RINGING状态时，如果应用程序调用SsmHangup或SsmHangupEx函数，会导致驱动程序向对端交换机发送呼叫拒绝消息。呼叫拒绝消息可以是CBK或CFL消息，具体使用哪一个，由本配置项决定。
注意事项	本配置项要求驱动程序的版本为 4.7.1.5 或更高。

3.1.2.27.7.7.2 来话呼叫：定制 ACM 消息

配置项	**DefaultACM**
节	**[TUP]**
书写格式	DefaultACM=0xab
取值范围	ab：8 比特数据，以 16 进制表示，8 个比特从高到低表示为 **HGFEDCBA**： **比特 / 含义 / 取值 / 描述** **HG / 备用** **F / 信令通道表示语 / 0 / 任何通道** **1 / 全部是 SS7 信令方式通道** **E / 呼叫前转表示语 / 0 / 呼叫不前转** **1 / 呼叫前转** **D / 来话回声抑制表示语 / 0 / 未包括来话半回声抑制器** **1 / 包括来话半回声抑制器** **C / 用户闲表示语 / 0 / 未指示** **1 / 用户闲** **BA / 地址全信号类型表示语 / 00 / 地址全信号** **01 / 地址全信号、计费** **10 / 地址全信号、免费** **11 / 地址全信号、投币式电话** 缺省值为 0x05。
功能描述	来话呼叫时，当本端完成被叫号码等信息的接收后，需要向对端交换机发送ACM消息，在ACM消息中应该包含指示被叫用户状态的消息表示语字段，本配置项用于设置此字段。更多信息请参加第 1 章中“TUP通道的状态转移”部分内容。
使用范例	DefaultACM=0x05

3.1.2.27.7.7.3 来话呼叫：定制 GRQ 消息

配置项	**ReqTypeIndicators**
节	**[TUP]**
书写格式	ReqTypeIndicators=0xk

<table>
<tr><td>取值范围</td><td>k：8 比特数据，以 16 进制表示，8 个比特从高到低表示为 HGFEDCBA：
<table>
<tr><th>比特</th><th>含义</th><th>取值</th><th>描述</th></tr>
<tr><td>HG</td><td>备用</td><td></td><td></td></tr>
<tr><td rowspan="2">F</td><td rowspan="2">回声抑制器请求表示语
(也适用于回声消除器)</td><td>0</td><td>未请求</td></tr>
<tr><td>1</td><td>请求</td></tr>
<tr><td rowspan="2">E</td><td rowspan="2">请求保持表示语</td><td>0</td><td>未请求</td></tr>
<tr><td>1</td><td>请求</td></tr>
<tr><td rowspan="2">D</td><td rowspan="2">恶意呼叫识别表示语</td><td>0</td><td>未遇到</td></tr>
<tr><td>1</td><td>遇到</td></tr>
<tr><td rowspan="2">C</td><td rowspan="2">原被叫地址表示语</td><td>0</td><td>未请求</td></tr>
<tr><td>1</td><td>请求</td></tr>
<tr><td rowspan="2">B</td><td rowspan="2">主叫用户线标识表示语</td><td>0</td><td>未请求</td></tr>
<tr><td>1</td><td>请求</td></tr>
<tr><td rowspan="2">A</td><td rowspan="2">主叫用户类别表示语</td><td>0</td><td>未请求</td></tr>
<tr><td>1</td><td>请求</td></tr>
</table>
缺省值为 0x03。</td></tr>
<tr><td>功能描述</td><td>设置 GRQ 消息中的请求类型指示语。</td></tr>
<tr><td>使用范例</td><td>ReqTypeIndicators=0x03</td></tr>
</table>

3.1.2.27.7.7.4 去话呼叫：定制 IAI/IAM 消息

配置项	**ConnectReqMsg**
节	**[TUP]**
书写格式	ConnectReqMsg=m
取值范围	m=0：由驱动程序自动选择。如果驱动程序内部的去话主叫号码缓冲区（可通过函数 SsmSetTxCallerId 或配置项 TxCallerId 设置）或原被叫号码缓冲区（可通过函数 SsmSetTxOriginalCallerID设置）不为空，使用IAI消息；否则使用IAM消息； m=1：使用 IAM 消息； m=2：使用 IAI 消息。 缺省值为 0。
功能描述	本端发起去话呼叫时，本配置项设置使用的初始地址消息的类型。

配置项	**CalloutIAM_CAT**
节	**[TUP]**
书写格式	CalloutIAM_CAT=0xk

<table>
<tr><td>取值范围</td><td>k：8 比特数据，以 16 进制表示，8 个比特从高到低表示为 HGFEDCBA：
<table>
<tr><th colspan="4">低位字节：HGFEDCBA</th></tr>
<tr><th>比特</th><th>含义</th><th>取值</th><th>描述</th></tr>
<tr><td>HG</td><td>备用</td><td>00</td><td></td></tr>
<tr><td rowspan="32">FEDCBA</td><td rowspan="32">主叫用户类别</td><td>000000</td><td>来源未知，在国际半自动接续中使用</td></tr>
<tr><td>000001</td><td>话务员，法语，在国际半自动接续中使用</td></tr>
<tr><td>000010</td><td>话务员，英语，在国际半自动接续中使用</td></tr>
<tr><td>000011</td><td>话务员，德语，在国际半自动接续中使用</td></tr>
<tr><td>000100</td><td>话务员，俄语，在国际半自动接续中使用</td></tr>
<tr><td>000101</td><td>话务员，西班牙语，在国际半自动接续中使用</td></tr>
<tr><td>000110</td><td>双方协商采用的语言(汉语)，在国际半自动接续中使用</td></tr>
<tr><td>000111</td><td>双方协商采用的语言，在国际半自动接续中使用</td></tr>
<tr><td>001000</td><td>双方协商采用的语言(日语)，在国际半自动接续中使用</td></tr>
<tr><td>001001</td><td>国内话务员（具有插入性能）</td></tr>
<tr><td>001010</td><td>普通用户、在长（国际）--长，长（国际）--市，局间使用</td></tr>
<tr><td>001011</td><td>优先用户、在长（国际）--长，长（国际）--市，市--市局间使用</td></tr>
<tr><td>001100</td><td>数据呼叫</td></tr>
<tr><td>001101</td><td>测试呼叫</td></tr>
<tr><td>001110</td><td>至 001111 备用</td></tr>
<tr><td>010000</td><td>普通、免费，在市--长（国际）局间使用</td></tr>
<tr><td>010001</td><td>普通、定期，在市--长（国际）局间使用</td></tr>
<tr><td>010010</td><td>普通、用户表、立即，在市--长（国际）局间使用</td></tr>
<tr><td>010011</td><td>普通、打印机、立即，在市--长（国际）局间使用</td></tr>
<tr><td>010100</td><td>优先、免费，在市--长（国际）局间使用</td></tr>
<tr><td>010101</td><td>优先、定期，在市--长（国际）局间使用</td></tr>
<tr><td>010110</td><td>备用</td></tr>
<tr><td>010111</td><td>备用</td></tr>
<tr><td>011000</td><td>普通用户，在市--市局间使用</td></tr>
<tr><td>011001</td><td>至 111111 备用</td></tr>
</table>
缺省值为 0x18。</td></tr>
<tr><td>功能描述</td><td>设置 IAI/IAM 消息中的主叫用户类别字段。</td></tr>
<tr><td>使用范例</td><td>CalloutIAM_CAT=0x18</td></tr>
</table>

<table>
<tr><td>配置项</td><td>CalloutIAM_MsgPntr</td></tr>
<tr><td>节</td><td>[TUP]</td></tr>
<tr><td>书写格式</td><td>CalloutIAM_MsgPntr=0xabcd</td></tr>
<tr><td>取值范围</td><td>abcd：16 比特数据，以 16 进制表示，低 12 比特有效，Bit11~Bit0 依次记为 LKJIHGFEDCBA，各比特的含义为：
<table>
<tr><th>比特</th><th>含义</th><th>取值</th><th>描述</th></tr>
<tr><td>H</td><td>国际来话呼叫表示语</td><td>0</td><td>不是国际来话呼叫</td></tr>
<tr><td></td><td></td><td>1</td><td>是国际来话呼叫</td></tr>
<tr><td>G</td><td>去话回声抑制器表示语</td><td>0</td><td>未包括去话回声抑制器</td></tr>
<tr><td></td><td></td><td>1</td><td>包括了去话回声抑制器</td></tr>
<tr><td rowspan="4">FE</td><td rowspan="4">导通检验表示语</td><td>00</td><td>不需要进行导通检验</td></tr>
<tr><td>01</td><td>该段电路需要进行导通检验</td></tr>
<tr><td>10</td><td>在前段电路进行了导通检验</td></tr>
<tr><td>11</td><td>备用</td></tr>
<tr><td rowspan="3">DC</td><td rowspan="3">电路性质表示语</td><td>00</td><td>在接续中无卫星电路</td></tr>
<tr><td>01</td><td>在接续中有卫星电路</td></tr>
<tr><td>其它</td><td>备用</td></tr>
<tr><td rowspan="4">BA</td><td rowspan="4">地址性质表示语</td><td>00</td><td>市话用户号码</td></tr>
<tr><td>01</td><td>备用</td></tr>
<tr><td>10</td><td>国内有效号码</td></tr>
<tr><td>11</td><td>国际号码</td></tr>
<tr><td>L</td><td>备用</td><td></td><td></td></tr>
<tr><td>K</td><td>信号通道表示语</td><td>0</td><td>任何通道</td></tr>
<tr><td></td><td></td><td>1</td><td>全部是 N0.7 信号方式通道</td></tr>
<tr><td>J</td><td>需要全部是数字通道表示语</td><td>0</td><td>普通呼叫</td></tr>
<tr><td></td><td></td><td>1</td><td>需要全数字通道</td></tr>
<tr><td>I</td><td>改发呼叫表示语</td><td>0</td><td>非改发呼叫</td></tr>
<tr><td></td><td></td><td>1</td><td>改发呼叫</td></tr>
</table>
缺省值为 0x0000。</td></tr>
</table>

功能描述	设置 IAI/IAM 消息中的消息表示语字段。
使用范例	CalloutIAM_MsgPntr=0x0000

配置项	**CallingIndicatorBit**
节	**[TUP]**
书写格式	CallingIndicatorBit=n
取值范围	n=0x10：使用比特 E； n=0x04：使用比特 C（缺省值）。
功能描述	在 IAI 消息中，有一个第 1 表示语八位位组(octet)，从 Bit7~Bit0 依次记为 HGFEDCBA，其中，比特 C 为附加主叫用户信息表示语，比特 E 为主叫用户线标识表示语。如果在 IAI 消息中包含了主叫号码，有的交换机要求使用比特 E，有的交换机要求使用比特 C。具体使用哪一个，应根据对端交换机的实际需要，通过本配置项进行设置。
使用范例	CallingIndicatorBit=0x10

配置项	**OriginalCalleeAddrInd**
节	**[TUP]**
书写格式	OriginalCalleeAddrInd=0xn
取值范围	n：16 进制的地址表示语，共 8 比特，低 4 比特有效。从 Bit3~Bit0，依次记为 DCBA，各比特的含义为： **比特** / **含义** / **取值** / **描述** **DC** / 备用 / 00 / **BA** / 地址性质表示语 / 00 / 市内用户号码 01 / 国内备用 10 / 国内有效号码 11 / 国际号码 缺省值为 0x02
功能描述	设置 IAI 消息中原被叫地址字段的地址表示语。
使用范例	OriginalCalleeAddrInd=0x02

3.1.2.27.7.7.5 去话呼叫：设置应答 GRQ 消息的模式

配置项	**AutoSendGSM**
节	**[TUP]**
书写格式	AutoSendGSM=m
取值范围	m=0：由应用程序接管，通道迁移到“挂起”状态，应用程序调用SsmSetTxCallerId设置主叫号码后恢复去话呼叫过程； m=1：由驱动程序自动应答（缺省值）。
功能描述	去话呼叫时，如果收到对端交换机请求主叫号码的GRQ消息，本配置项用于决定是否由驱动程序自动应答GSM消息。更多信息请参见第 1 章中“TUP通道的状态转移”部分内容。

3.1.2.27.7.7.6 不使用 SynCTI 提供的 TUP 协议状态机

配置项	**AutoHandleTup**
节	[SS7]
书写格式	AutoHandleTup=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否使用 SynCTI 驱动程序提供的 TUP 协议状态机。

3.1.2.27.7.7.7 输出调试信息

配置项	**DebugViewTupCallProc**
节	**[SS7]**
书写格式	DebugViewTupCallProc=b

取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否将 TUP 协议状态机的调试信息输出到 DebugView 工具软件中。如果选择是，需要先运行 DebugView 工具软件。
注意事项	开启 TUP 协议的调试功能后，会降低应用系统的运行效率，因此仅用于调试的目的。

3.1.2.27.7.8 ISUP 协议高级配置项

3.1.2.27.7.8.1 来话呼叫：定制本端接受来话呼叫时应答的消息类型

配置项	**DefaultCalledPickupMsg**
节	**[ISUP]**
书写格式	DefaultCalledPickupMsg=k
取值范围	k=0x09：使用 ANM 消息（应答）； k=0x07：使用 CON 消息（地址全并摘机）。 缺省值为 0x09。
功能描述	当ISUP通道处于“振铃”状态时，如果应用程序调用SsmPickup或SsmPickupANX函数，本配置项设置向对端交换机发送的消息类型。
注意事项	本配置项要求 SynCTI Ver. 4.7.1.8 及更高。
使用范例	DefaultCalledPickupMsg=0x09

3.1.2.27.7.8.2 来话呼叫：定制后向呼叫指示码

配置项	**DefaultBackwardCallInd**
节	**[ISUP]**
书写格式	DefaultBackwardCallInd=k

<table>
<tr><td>取值范围</td><td>k：后向呼叫表示语字段的参数值，包含 2 个字节，从最高位比特（Bit15）到最低位比特（Bit0）记为 PONMLKJIHGFEDCBA，各比特含义如下：
<table>
<tr><th>比特组合</th><th>含义</th><th>取值范围及描述</th></tr>
<tr><td>BA</td><td>计费表示语</td><td>00：无指示
01：不计费
10：计费
11：备用</td></tr>
<tr><td>DC</td><td>被叫用户状态表示语</td><td>00：无指示
01：用户空
10：空时连接
11：备用</td></tr>
<tr><td>FE</td><td>被叫用户类别表示语</td><td>00：无指示
01：普通用户
10：付费电话
11：备用</td></tr>
<tr><td>HG</td><td>端至端方式表示语 (注)</td><td>00：端至端方式不可用 (只有逐段转发方式可用)
01：传递方式可用
10：SCCP 方式可用
11：传递方式和 SCCP 方式可用</td></tr>
<tr><td>I</td><td>互通表示语(注)</td><td>0：未遇到互通
1：遇到互通</td></tr>
<tr><td>J</td><td>端至端信息表示语(注)</td><td>0：无端至端信息可用
1：端至端信息可用</td></tr>
<tr><td>K</td><td>ISDN 用户部分表示语 (注)</td><td>0：不是所有方向都使用 ISDN 用户部分
1：所有方向都使用 ISDN 用户部分</td></tr>
<tr><td>L</td><td>保持表示语</td><td>0：未请求保持
1：请求保持</td></tr>
<tr><td>M</td><td>ISDN 接入表示语</td><td>0：终端接入非 ISDN
1：终端接入 ISDN</td></tr>
<tr><td>N</td><td>回声控制装置表示语</td><td>0：未包括来话半回声控制装置
1：包括来话半回声控制装置</td></tr>
<tr><td>PO</td><td>SCCP 方式表示语(注)</td><td>00：无指示
01：无连接方式可用
10：面向连接方式可用
11：无连接和面向连接方式可用</td></tr>
</table>
注：比特 G-K 和比特 O-P 组成协议控制表示语
缺省值为 0x1416。</td></tr>
<tr><td>功能描述</td><td>设置 ACM 和 CON 消息中的后向呼叫表示语字段。</td></tr>
<tr><td>注意事项</td><td>本配置项要求 SynCTI Ver. 4.7.1.8 及更高。</td></tr>
<tr><td>使用范例</td><td>DefaultBackwardCallInd=0x1416</td></tr>
</table>

3.1.2.27.7.8.3 来话呼叫：定制 REL 消息

<table>
<tr><td>配置项</td><td>DefaultHangupRELInd</td></tr>
<tr><td>节</td><td>[ISUP]</td></tr>
<tr><td>书写格式</td><td>DefaultHangupRELInd=0xk</td></tr>
<tr><td>取值范围</td><td>k 的取值范围及含义如下表所示：
<table>
<tr><th>取值</th><th>shpa3api.h 中的宏</th><th>描述</th></tr>
<tr><td>0x01</td><td>C_ISUP_REL_UNN</td><td>空号</td></tr>
<tr><td>0x10</td><td>C_ISUP_REL_NORMAL_REL</td><td>正常拆线</td></tr>
<tr><td>0x11</td><td>C_ISUP_REL_BUSY</td><td>用户忙</td></tr>
<tr><td>0x12</td><td>C_ISUP_REL_NOREPLY</td><td>无应答</td></tr>
<tr><td>0x15</td><td>C_ISUP_REL_DENY</td><td>拒绝</td></tr>
<tr><td>0x1c</td><td>C_ISUP_REL_LACKADDR</td><td>地址不全</td></tr>
<tr><td>0x1f</td><td>C_ISUP_REL_NORMAL</td><td>正常</td></tr>
<tr><td>0x2a</td><td>C_ISUP_REL_BLOCK</td><td>交换设备拥塞</td></tr>
</table>
缺省值为 0x15(C_ISUP_REL_DENY)。</td></tr>
<tr><td>功能描述</td><td>当应用程序调用SsmHangup函数而引发驱动程序向对端交换机发送REL消息时，本配置项设置REL消息中携带的释放原因值。</td></tr>
</table>

注意事项	● 如果设定值不在取值范围之内，驱动程序会自动调整为 0x15； ● 本配置项要求 SynCTI Ver. 4.7.1.5 或更高。
使用范例	DefaultHangupRELInd=0x15

配置项	**DefaultCauseInd**
节	**[ISUP]**
书写格式	DefaultCauseInd=n
取值范围	n：16 比特数据，以 16 进制表示，各比特的含义为： **低位字节：HGFEDCBA** **比特** / **含义** / **取值** / **描述** H / 扩充表示语 / 0 / 八位位组延续到下一个八位位组（建议八位位组） 1 / 最后的八位位组 GF / 编码标准 / 00 / ITU-T 标准编码 其它 / 保留 E / 备用 / 0 / 应设置为 0 DCBA / 位置 / 0000 / 用户（U） 0001 / 为本地用户提供服务的专用网（LPN） 0010 / 为本地用户提供服务的公用网（LN） 0011 / 转接网络（TN） 0100 / 为远端用户提供服务的公用网（RLN） 0101 / 为远端用户提供服务的专用网（RPN） 0111 / 国际网络（INTL） 1010 / 互通点以外的网路（BI） 其它 / 备用 **高位字节：HGFEDCBA** **比特** / **含义** / **取值** / **描述** H / 扩充表示语 / 1 / 最后的八位位组 GFEDCBA / 建议 / 0000000 / Q.931 0000011 / X.21 0000100 / X.25 0000101 / 公共陆地移动网，Q.1031/Q.1051 其它 / 保留 缺省值为 0x0000。
功能描述	设置 REL 消息中的原因表示语字段的编码标准位置及建议参数，其中建议参数可选。若不选择建议参数，则低 8 比特的扩充位必须置 1，驱动将忽略高 8 比特值；若选择建议参数，则低 8 比特的扩充位必须置 0。
使用范例	DefaultCauseInd=0x0000

3.1.2.27.7.8.4 来话呼叫：设置其它参数

配置项	**SaveRGNTo1stPhoNumStr**
节	**[ISUP]**
书写格式	SaveRGNTo1stPhoNumStr=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否可以通过函数SsmGet1stPhoNumStr或SsmGet1stPhoNumStrA获取IAM消息中的改发号码信息。更多信息请参见上述函数的说明。
注意事项	● 本配置项仅适用于 ISUP 通道； ● 本配置项要求 SynCTI 驱动程序的版本为 Ver 4.7.1.7 或更高。

3.1.2.27.7.8.5 去话呼叫：定制 IAM 消息

配置项	**DefaultNatureOfConnectionInd**
节	**[ISUP]**
书写格式	DefaultNatureOfConnectionInd=0xk
取值范围	k：16 进制形式的连接性质表示码，缺省值为 0x00。具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的连接性质表示语。

使用范例	DefaultNatureOfConnectionInd=0x00

配置项	**DefaultIAM_ForwardCallInd**
节	**[ISUP]**
书写格式	DefaultIAM_ForwardCallInd=0xk
取值范围	k：16 进制形式的前向呼叫指示语，缺省值为 0x0040。具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的前向呼叫指示语。
使用范例	DefaultIAM_ForwardCallInd=0x0040

配置项	**DefaultIAM_CAT**
节	**[ISUP]**
书写格式	DefaultIAM_CAT=0xk
取值范围	k：16 进制形式的主叫用户类别，缺省值为 0x0a。具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的主叫用户类别。
使用范例	DefaultIAM_CAT=0x0a

配置项	**DefaultIAM_TransmissionMediumRequirment**
节	**[ISUP]**
书写格式	DefaultIAM_TransmissionMediumRequirment=0xk
取值范围	k：16 进制形式的传输媒介要求，缺省值为 0x02。具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的传输媒介要求。
使用范例	DefaultIAM_TransmissionMediumRequirment=0x02

配置项	**DefaultIAM_CalleeParam**
节	**[ISUP]**
书写格式	DefaultIAM_CalleeParam=0xk
取值范围	k：16 比特数据，以 16 进制表示，各比特的含义为： **低位字节：HGFEDCBA** 比特 | 含义 | 取值 | 描述 H | 奇偶指示码 | 0 | 地址为偶数个 1 | 地址为奇数个 GFEDCBA | 地址性质指示码 | 0000001 | 用户号码 0000011 | 国内号码 0000100 | 国际号码 其它 | 备用 **高位字节：HGFEDCBA** 比特 | 含义 | 取值 | 描述 H | 内部网络号码指示码 | 0 | 允许路由到内部网络号码 1 | 不允许路由到内部网络号码 GFE | 号码计划指示码 | 001 | ISDN(电话)号码计划(建议 E.164，E.163) 011 | 数据号码计划（建议 X.164） 100 | 用户电报号码计划（建议 F.164） 其它 | 备用 DCBA | 备用 缺省值为 0x1003。
功能描述	设置 IAM 消息中被叫用户号码字段的被叫参数。
使用范例	DefaultIAM_CalleeParam=0x1003

配置项	**DefaultIAM_CallerParam**
节	**[ISUP]**
书写格式	DefaultIAM_CallerParam=0xk

<table>
<tr><td>取值范围</td><td>k：16 比特数据，以 16 进制表示，各比特的含义为：
<table>
<tr><td colspan="4">低位字节：HGFEDCBA</td></tr>
<tr><td>比特</td><td>含义</td><td>取值</td><td>描述</td></tr>
<tr><td rowspan="2">H</td><td rowspan="2">奇偶指示码</td><td>0</td><td>地址为偶数个</td></tr>
<tr><td>1</td><td>地址为奇数个</td></tr>
<tr><td rowspan="4">GFEDCBA</td><td rowspan="4">地址性质指示码</td><td>0000001</td><td>用户号码</td></tr>
<tr><td>0000011</td><td>国内号码</td></tr>
<tr><td>0000100</td><td>国际号码</td></tr>
<tr><td>其它</td><td>备用</td></tr>
<tr><td colspan="4">高位字节：HGFEDCB</td></tr>
<tr><td>比特</td><td>含义</td><td>取值</td><td>描述</td></tr>
<tr><td rowspan="2">H</td><td rowspan="2">主叫号码不全指示码</td><td>0</td><td>全</td></tr>
<tr><td>1</td><td>不全</td></tr>
<tr><td rowspan="4">GFE</td><td rowspan="4">号码计划指示码</td><td>001</td><td>ISDN(电话)号码计划(建议 E.164，E.163)</td></tr>
<tr><td>011</td><td>数据号码计划（建议 X.164）</td></tr>
<tr><td>100</td><td>用户电报号码计划（建议 F.164）</td></tr>
<tr><td>其它</td><td>备用</td></tr>
<tr><td rowspan="4">DC</td><td rowspan="4">地址提供限制指示码</td><td>00</td><td>不限制提供</td></tr>
<tr><td>01</td><td>限制提供</td></tr>
<tr><td>10</td><td>地址不可获得</td></tr>
<tr><td>其它</td><td>备用</td></tr>
<tr><td rowspan="4">BA</td><td rowspan="4">显示指示码</td><td>00</td><td>用户提供，不验证；（国内用）</td></tr>
<tr><td>01</td><td>用户保留，验证和通过</td></tr>
<tr><td>10</td><td>用户提供，验证和失败；（国内用）</td></tr>
<tr><td>11</td><td>网络提供</td></tr>
</table>
缺省值为 0x1001</td></tr>
<tr><td>功能描述</td><td>设置 IAM 消息中主叫用户号码字段的主叫参数。</td></tr>
<tr><td>使用范例</td><td>DefaultIAM_CallerParam=0x1001</td></tr>
</table>

配置项	**DefaultIAM_OriginalCalleeParam**
节	**[ISUP]**
书写格式	DefaultIAM_OriginalCalleeParam=0xk
取值范围	k：16 进制形式的 16 比特数据，缺省值为 0x1001。具体取值请参见 ISUP 协议的相关说明。
功能描述	设置 IAM 消息中原被叫的头两字节信息，包括地址性质表示语、编号计划、限制提供。具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
使用范例	DefaultIAM_OriginalCalleeParam=0x1001

配置项	**bSubscriberSI** **SubscriberSI**
节	**[ISUP]**
书写格式	bSubscriberSI=k SubscriberSI=0xk_1, 0xk_2,…0xk_i,…, 0xk_{11}
取值范围	k：=0：不包含（缺省值）； =1：包含，用户业务信息的具体值在配置项 SubscriberSI 中指定。 k_i：16 进制形式的用户业务信息，最多可以配置 11 个，缺省值为 0x80,0x90,0xa3，适用于华为交换机。k_i 的具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
功能描述	bSubscriberSI 设置 IAM 消息中是否包含用户业务信息内容； SubscriberSI 设置用户业务信息的内容，只有当 bSubscriberSI 设置为 1 时才有效。
使用范例	bSubscriberSI=1 SubscriberSI=0x80,0x90,0xa3

配置项	**bOptionalFCI** **OptionalFCI**
节	**[ISUP]**

<table>
<tr><td>书写格式</td><td>bOptionalFCI=b
OptionalFCI=0xk</td></tr>
<tr><td>取值范围</td><td>b：=0：不包含（缺省值）；
=1：包含。
k：为 16 进制形式的 8 比特数据，按照从高比特位到低比特位的顺序记为 HGFEDCBA，各比特的含义为：
<table>
<tr><th>比特</th><th>含义</th><th>取值</th><th>描述</th></tr>
<tr><td rowspan="2">H</td><td rowspan="2">被连接线识别请求表示语</td><td>0</td><td>未请求</td></tr>
<tr><td>1</td><td>请求</td></tr>
<tr><td>GFED</td><td>备用</td><td></td><td></td></tr>
<tr><td rowspan="2">C</td><td rowspan="2">简单分段表示语</td><td>0</td><td>将不发送附加的信息</td></tr>
<tr><td>1</td><td>将在分段消息中发送附加的信息</td></tr>
<tr><td rowspan="4">BA</td><td rowspan="4">闭合用户群呼叫表示语</td><td>00</td><td>非 CUG 呼叫</td></tr>
<tr><td>01</td><td>备用</td></tr>
<tr><td>10</td><td>闭合用户群呼叫，允许出接口</td></tr>
<tr><td>11</td><td>闭合用户群呼叫，不允许出接口</td></tr>
</table>
缺省值为 0。如果需要配置，通常由对端交换机根据其具体配置提供实际数值。</td></tr>
<tr><td>功能描述</td><td>bOptionalFCI 设置 IAM 消息中是否包含任选前向呼叫表示语。如果包含，任选前向呼叫表示语的值在配置项 OptionalFCI 中指定。
OptionalFCI 任选前向呼叫表示语的具体值。</td></tr>
<tr><td>使用范例</td><td>bOptionalFCI=1
OptionalFCI=0x01</td></tr>
</table>

配置项	**Usr2UsrInfo**
节	**[ISUP]**
书写格式	Usr2UsrInfo=s_1,s_2,…,s_i,…,s_N
取值范围	s_i：十六进制字符，最多可以设置 131 个（即 N 最大为 131）。缺省值为空，即不包含该字段。
功能描述	设置 IAM 消息中的用户至用户信息字段。本配置项设置的功能也可以通过函数 SsmSetIsupParameter进行设置。
使用范例	Usr2UsrInfo=a0,31,6c

3.1.2.27.7.8.6 去话呼叫：设置应答 INF 消息的模式

配置项	**AutoSendINF**
节	**[ISUP]**
书写格式	AutoSendINF=m
取值范围	m=0：由应用程序接管； m=1：由驱动程序自动应答（缺省值）。
功能描述	去话呼叫时，如果收到对端交换机请求主叫号码的INR消息，本配置项用于决定是否由驱动程序自动应答INF消息。详细信息请参见第 1 章中“ISUP通道的状态转移”部分内容。

3.1.2.27.7.8.7 不使用 SynCTI 提供的 ISUP 协议状态机

配置项	**AutoHandleIsup**
节	[SS7]
书写格式	AutoHandleIsup=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否使用 SynCTI 驱动程序提供的 ISUP 协议状态机。

3.1.2.27.7.9 SCCP 协议的高级配置项

配置项	**AutoHandleSccp**
节	**[SS7]**
书写格式	AutoHandleSccp=b

取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否由 SynCTI 驱动程序处理 SCCP 消息。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 4.7.1.7 或更高。

3.1.2.27.8ISDN 信令的高级设置

3.1.2.27.8.1 设置接续模式

配置项	**AutoHandleIsdn**
节	**[ISDN]**
书写格式	AutoHandleIsdn=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置是否由驱动程序提供的ISDN状态机来处理ISDN消息。更多信息请参见第 1 章中“ISDN高级编程接口”部分内容。
注意事项	本配置项为高级配置，只有当应用程序希望自行处理 ISDN 消息时，才需要设置本配置项。

3.1.2.27.8.2 设置用户侧/网络侧数字中继线的工作参数

配置项	**UserCrcMode** **NetCrcMode**
节	**[ISDN]**
书写格式	UserCrcMode[n]=b NetCrcMode[n]=b
取值范围	n：用户侧（对于 UserCrcMode）或网络侧（对于 NetCrcMode）数字中继线的编号。 b：CRC 校验功能开关， b=0：关闭； b=1：开启。
功能描述	设置是否开启数字中继线上的 CRC 校验功能。UserCrcMode 适用于用户侧，NetCrcMode 适用于网络侧。
注意事项	● 本配置项为高级配置； ● 如果配置项 TotalUserLinker 的设定值(假定为 N)大于 0，UserCrcMode 必须设置 N 次；如果配置项 TotalNetLinker 的设定值(假定为 N)大于 0，NetCrcMode 必须设置 N 次。

配置项	**UserChIdentify** **NetChIdentify**
节	**[ISDN]**
书写格式	UserChIdentify[n]=k NetChIdentify[n]=k
取值范围	n：用户侧（对于 UserChIdentify）或网络侧（对于 NetChIdentify）数字中继线的编号； k：通路识别表示方法， k=0x11：时隙图； k=0x10：号码(缺省值)。 **注意：k 的缺省值取决于 SynCTI 驱动程序的版本：如果版本低于 Ver. 4.7.3.1，k 的缺省值为 0x11；否则为 0x10。**
功能描述	设置数字中继线上的通路识别表示方法。UserChIdentify适用于用户侧，NetChIdentify适用于网络侧。有关通路识别表示方法的更多信息请参见第 1 章中“通路识别表示方法”部分内容。

配置项	**UserVoiceFormat** **NetVoiceFormat**
节	**[ISDN]**

书写格式	UserVoiceFormat[n]=k NetVoiceFormat[n]=k
取值范围	n：用户侧（UserVoiceFormat）或网络侧（NetVoiceFormat）数字中继线的编号。 k：语音编码格式， k=6：A-law（缺省值）； k=7：μ-law。
功能描述	设置数字中继线上语音信道的语音编码格式。 UserVoiceFormat 适用于用户侧，NetVoiceFormat 适用于网络侧。

配置项	**UserRestarTime** **NetRestarTime**
节	**[ISDN]**
书写格式	UserRestarTime=t NetRestarTime=t
取值范围	100≤t≤120，单位为秒，缺省值为 100。
功能描述	设置本端响应对端链路重启命令的时间，即从本端链路收到对端的重启命令开始，到完成重启动作的时间。如果在本配置项设定的时间内，链路未能完成重启，则相关的数字中继线上全部通道将进入不可用状态（即 S_CALL_UNAVAILABLE）。 UserRestarTime 适用于用户侧，NetRestarTime 适用于网络侧。
注意事项	● 本配置项为高级配置项，大多数情况下无需进行设置； ● 本配置项对系统中所有数字中继线都起作用。

配置项	**UserEstablishTime** **NetEstablishTime**
节	**[ISDN]**
书写格式	UserEstablishTime=t NetEstablishTime=t
取值范围	5≤t≤10，单位为秒，缺省值为 5。
功能描述	设置驱动程序尝试将本端与对端建立链路连接的时间间隔。如果不能建立链路连接，在本配置项设置的时间后，驱动程序会自动开始下一次尝试。 UserEstablishTime 适用于用户侧，NetEstablishTime 适用于网络侧。
注意事项	● 本配置项为高级配置项，大多数情况下无需进行设置； ● 本配置项对系统中所有的用户侧或网络侧数字中继线都起作用。

3.1.2.27.8.3 设置去话呼叫的工作参数

配置项	**UserCalledNoSet** **NetCalledNoSet** **UserCallingNoSet** **NetCallingNoSet**
节	**[ISDN]**
书写格式	UserCalledNoSet[n]=k NetCalledNoSet[n]=k UserCallingNoSet[n]=b NetCallingNoSet[n]=g

取值范围	n：用户侧（对于 UserCalledNoSet、UserCallingNoSet）或网络侧（对于 NetCalledNoSet、NetCallingNoSet）数字中继线的编号。 k：号码类型及编号方案，16 进制形式， k=0xa1：国内号码（缺省值）； k=0x91：国际号码； k=0xc1：用户号码； k=0xb1：特定号码； k=0x81：未知。 g：号码类型及编号方案，16 进制形式， g=0x21：国内号码（缺省值）； g=0x11：国际号码； g=0x41：用户号码； g=0x31：特定号码； g=0x01：未知。
功能描述	去话呼叫时，设置 setup 消息中主叫号码或被叫号码的号码类型及编号方案。 UserCalledNoSet、NetCalledNoSet 分别用于设置用户侧、网络侧的被叫号码，UserCallingNoSet、NetCallingNoSet 分别用于设置用户侧、网络侧的主叫号码。
注意事项	● 本配置项为高级配置； ● 如果配置项 TotalUserLinker 的设定值(假定为 N)大于 0，UserCalledNoSet 和 UserCallingNoSet 必须设置 N 次；如果配置项 TotalNetLinker 的设定值(假定为 N)大于 0，NetCalledNoSet 和 NetCallingNoSet 必须设置 N 次。

配置项	**UserNumIsFull** **NetNumIsFull**
节	**[ISDN]**
书写格式	UserNumIsFull=b NetNumIsFull=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	去话呼叫时，是否在 SETUP 消息中包含“被叫号码全”参数。 UserNumIsFull 适用于用户侧，NetNumIsFull 适用于网络侧。
注意事项	● 本配置项为高级配置。

配置项	**UserChPreference** **NetChPreference**
节	**[ISDN]**
书写格式	UserChPreference=b NetChPreference=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	是否需要优选通路类型。 UserChPreference 适用于用户侧，NetChPreference 适用于网络侧。
注意事项	● 本配置项为高级配置。

配置项	**UserHighLayerCompatible** **NetHighLayerCompatible**
节	**[ISDN]**
书写格式	UserHighLayerCompatible=b NetHighLayerCompatible=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置 SETUP 消息中是否包含“高层兼容性”字段。 UserHighLayerCompatible 适用于用户侧，NetHighLayerCompatible 适用于网络侧。

注意事项	● 本配置项为高级配置。

配置项	**UserLowLayerCompatible** **NetLowLayerCompatible**
节	**[ISDN]**
书写格式	UserLowLayerCompatible=b NetLowLayerCompatible=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	设置 SETUP 消息中是否包含“低层兼容性”字段。 UserLowLayerCompatible 适用于用户侧，NetLowLayerCompatible 适用于网络侧。
注意事项	● 本配置项为高级配置。

配置项	**UserDialTime** **NetDialTime**
节	**[ISDN]**
书写格式	UserDialTime=t NetDialTime=t
取值范围	t：等待时间，单位为秒，缺省值为 60 秒。
功能描述	当应用程序在某个ISDN通道上调用函数SsmPickup或SsmSearchIdleCallOutCh后，驱动程序会启动一个定时器，应用程序应当在定时器溢出前开始去话呼叫。如果定时器溢出，驱动程序将自动回收该通道，将其迁移到S_CALL_STANDBY状态。本配置项用于设置此定时器的值。更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。 UserDialTime 适用于用户侧，NetDialTime 适用于网络侧。
注意事项	● 本配置项为高级配置。

配置项	**TransferCapability**
节	**[ISDN]**
书写格式	**TransferCapability=n**
取值范围	=0：语音，默认 =1：3.1k 音频
功能描述	设置信令消息中“负载能力”字段的内容。
注意事项	● 本配置项为高级配置。

配置项	**PresentNumber**
节	**[ISDN]**
书写格式	PresentNumber =n
取值范围	=0：不允许显示号码 =1：允许显示号码，默认
功能描述	设置信令消息中主叫号码是否“允许显示”字段的内容。
注意事项	● 本配置项为高级配置。

配置项	**UserT303** **NetT303**
节	**[ISDN]**
书写格式	UserT303=t NetT303=t
取值范围	t：定时器的时长，单位为秒，缺省值为 4 秒。
功能描述	设置定时器T303。UserT303 适用于用户侧，NetT303 适用于网络侧。有关T303 的更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。

配置项	**UserT304** **NetT304**
节	**[SS7]**
书写格式	UserT304=t NetT304=t
取值范围	t：定时器的时长，单位为秒，缺省值为 30 秒。
功能描述	设置定时器T304。UserT304 适用于用户侧，NetT304 适用于网络侧。有关T304 的更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。

配置项	**UserT310** **NetT310**
节	**[ISDN]**
书写格式	UserT310=t NetT310=t
取值范围	t：定时器的时长，单位为秒，缺省值为 10 秒。
功能描述	设置定时器T310。UserT310 适用于用户侧，NetT310 适用于网络侧。有关T310 的更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。

3.1.2.27.8.4 设置来话呼叫的工作参数

配置项	**UserSideDefaultAckTimer** **UserSideDefaultAck** **NetSideDefaultAckTimer** **NetSideDefaultAck**
节	**[ISDN]**
书写格式	UserSideDefaultAckTimer=t UserSideDefaultAck=k NetSideDefaultAckTimer=t NetSideDefaultAck=k
取值范围	t： 定时器的设定值，单位为秒，t<120，缺省值为 20。 k：设置定时器溢出后驱动程序的动作，其取值及其含义如下表所示： K / 含义 / 驱动程序动作 1 / 被叫用户空闲 / 发送 ALERTING 消息 2 / 被叫用户忙 / 发送 DISCONNECT 消息，原因=被叫用户忙 4 / 呼叫拒绝 / 发送 DISCONNECT 消息，原因=呼叫拒绝 5 / 无应答 / 发送 DISCONNECT 消息，原因=被叫用户无应答 其它 / 备用 k 的缺省值为 1。
功能描述	来话呼叫时，当驱动程序完成全部号码的接收后，本配置项设置是否由驱动程序自动发送应答消息。 更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。 UserSideDefaultAckTimer、UserSideDefaultAck 适用于用户侧，NetSideDefaultAckTimer、NetSideDefaultAck 适用于网络侧。
注意事项	● 本配置项为高级配置

配置项	**UserIsReceivePhoNum** **NetIsReceivePhoNum**
节	**[ISDN]**
书写格式	UserIsReceivePhoNum=b NetIsReceivePhoNum=b
取值范围	b=0：汇接局方式； b=1：端局方式（缺省值）。
功能描述	设置来话呼叫时，驱动程序的工作方式。更多信息请参见第 1 章中“端局方式与汇接局方式”部分内容。 UserIsReceivePhoNum 适用于用户侧，NetIsReceivePhoNum 适用于网络侧。

注意事项	● 如果对端交换机为阿尔卡特或 AVAYA，通常应将本配置项设置为 0； ● 较早版本的驱动程序支持的配置项 UserTieStationMode 具有相同功能，但只支持用户侧，可以用 UserIsReceivePhoNum 来替代。

配置项	**UserIsSendChIdentify** **NetIsSendChIdentify**
节	**[ISDN]**
书写格式	UserIsSendChIdentify=b NetIsSendChIdentify=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	来话呼叫时，当本端收到对端交换机的 SETUP 消息后，在相应的应答信号中（如 CALL PROCEEDING、ALERT 等）是否包含通路识别信息。 UserIsSendChIdentify 适用于用户侧，NetIsSendChIdentify 适用于网络侧。
注意事项	如果对端交换机为西门子交换机，通常将本配置项设置为 1。

配置项	**UserT302** **NetT302**
节	**[ISDN]**
书写格式	UserT302=t NetT302=t
取值范围	t：定时器的时长，单位为秒，缺省值为 15 秒。
功能描述	设置定时器T302。UserT302 适用于用户侧，NetT302 适用于网络侧。有关T302 的更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。

配置项	**UserT313**
节	**[ISDN]**
书写格式	UserT313=t
取值范围	t：定时器的时长，单位为秒，缺省值为 4 秒。
功能描述	设置定时器T313。有关T313 的更多信息请参见第 1 章中“ISDN通道的状态转移图”部分内容。
注意事项	本配置项只适用于用户侧。

3.1.2.27.8.5 设置定时器

配置项	**UserT305** **NetT305** **NetT306**
节	**[ISDN]**
书写格式	UserT305=t NetT305=t NetT306=t
取值范围	t：定时器的时长，单位为秒，缺省值为 30 秒。
功能描述	设置定时器 T305 和 T306。UserT305 适用于用户侧，NetT305、NetT306 适用于网络侧。 ✧ 对于用户侧，当本端向对端交换机发送 DISCONNECT 消息后，驱动程序会启动 T305；当本端收到对端交换机的 RELEASE 或 DISCONNECT 消息后，驱动程序会自动关闭 T305。 ✧ 对于网络侧，当本端向对端交换机发送 DISCONNECT 消息后，如果 DISCONNECT 消息中包含了进展表示语字段，并且进展表示语字段的值等于 8，驱动程序会启动 T306，否则启动 T305。当驱动程序收到对端交换机的 RELEASE 或 DISCONNECT 消息后，会自动关闭 T305 或 T306。 如果 T305 或 T306 溢出，驱动程序会发送 RELEASE 消息。

配置项	**UserT308** **NetT308**
节	**[ISDN]**

书写格式	UserT308=t NetT308=t
取值范围	t：定时器的时长，单位为秒，缺省值为 4 秒。
功能描述	设置定时器 T308。UserT308 适用于用户侧，NetT308 适用于网络侧。 当本端向对端交换机发送 RELEASE 消息后，驱动程序会启动 T308；当本端收到对端交换机的 RELEASE 或 RELEASE COMPLETE 消息后，驱动程序会自动关闭 T308。如果 T308 溢出，驱动程序会重发 RELEASE 消息。

3.1.2.27.8.6 设置其它参数

配置项	**WaitHangupTime**
节	**[ISDN]**
书写格式	WaitHangupTime=b
取值范围	20<b<10000，缺省值为 1600，单位：毫秒。
功能描述	设置 ISDN 挂起等待时间。
注意事项	本配置项为高级配置。

配置项	**UserStatusReason** **NetStatusReason**
节	**[ISDN]**
书写格式	UserStatusReason=m NetStatusReason=m
取值范围	m=1：3 字节格式（缺省值）； m=0：2 字节格式 。
功能描述	当本端收到对端交换机发送的 STATUS ENQUIRY 消息（状态查询）时，驱动程序会自动应答 STATUS 消息。本配置项设置 STATUS 消息中的原因值字段是采用 2 字节格式还是 3 字节格式。具体采用哪一种格式，需要根据对端交换机的参数进行设置。UserStatusReason 适用于用户侧，NetStatusReason 适用于网络侧。
注意事项	● 本配置项为高级选项； ● 本配置项要求 SynCTI 驱动程序的版本为 Ver. 4.7.1.8 或更高。

3.1.2.27.8.7 输出调试信息

配置项	**DebugSwitch**
节	**[ISDN]**
书写格式	DebugSwitch=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否输出 ISDN 信令的调试信息。驱动程序提供了一个消息输出功能，可以将接收或发送的 ISDN 消息输出。开发人员可以通过工具软件 DebugView 对输出结果进行观测。
注意事项	本配置项为高级配置。

配置项	**RecordIFrame**
节	**[ISDN]**
书写格式	RecordIFrame=m
取值范围	m=0：不输出（缺省值）； m=1：输出到日志文件（*.log）； m=2：输出到 dbgview。
功能描述	设置驱动程序输出 ISDN 协议数据链路层 1 帧消息的调试信息的方式。
注意事项	本配置项为高级配置。

配置项	**DecodeUserSideIsdnMsg** **DecodeNetSideIsdnMsg**
节	**[ISDN]**

书写格式	DecodeUserSideIsdnMsg=b DecodeNetSideIsdnMsg=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否由驱动程序在输出 ISDN 调试消息前先将 ISDN 消息进行解码。 原始的 ISDN 消息是二进制形式，不方便阅读。解码后的 ISDN 消息是文本形式，可以直接阅读。 DecodeUserSideIsdnMsg 适用于用户侧，DecodeNetSideIsdnMsg 适用于网络侧。

配置项	**LogMode**
节	**[ISDN]**
书写格式	LogMode=n
取值范围	n=0：每 10 万条重新保存；（缺省值） n=1：每天保存一个日志。
功能描述	设置 ISDN 日志输出方式： n＝0 时，每 1 万条日志则从头开始覆盖先前的日志，每 10 万条日志则清空日志文件重新写； n＝1 时，根据系统的日期生成日志的文件名，并保存当天的日志。

3.1.2.27.9 设置 DSP 芯片的工作方式

配置项	**LoadFskBin**
节	**[BoardId=x]**
书写格式	LoadFskBin=m
取值范围	m=0：增强的回波抵消器方式。语音通道不再具有录音功能和 FSK 收发器功能，闲置出来的 DSP 资源被用来增强回波抵消器的效果； m=1：正常运行方式（缺省值）； m=2：16 时隙录音方式。驱动程序将 16 时隙上的原始信令数据通过录音的方式直接传递给应用程序进行处理。当 m=1 时，板卡上的每片 DSP 芯片处理 15 个语音时隙，每条数字中继线上 30 个时隙分别由 2 片 DSP 芯片进行处理，一片处理 TS_1~TS_{15}，另外一片处理 TS_{17}~TS_{31}。当 m=2 时，原来处理时隙 TS_{17}~TS_{31} 的 DSP 芯片转而处理时隙 TS_{16}~TS_{30}，TS_{31} 上的语音数据将被抛弃。**注意：**通道的物理编号 15 所对应的时隙不再是 TS_{17}，而是 TS_{16}，其余通道依此类推。
功能描述	选择 SHD 系列板卡上 DSP 芯片的工作程序。
注意事项	● 本配置项是高级配置项，只适用于 30 通道、60 通道和 120 通道规格的 SHD 系列板卡。

3.1.2.28 SHN 系列板卡常用配置项(CTI 系列)

3.1.2.28.1 SIP 信令的高级设置

3.1.2.28.1.1 设置 SIP 信令的工作参数

配置项	**LocalPort** **RTPRange** **AudioCodecList**
节	**[SIP]**
书写格式	LocalPort=n RTPRange= t_L, t_H AudioCodecList= p_1,p_2,…. p_n

取值范围	m：设置 SIP 信令承载的网络协议，具体为： m=0：使用 UDP 协议承载，默认值； m=1：使用 TCP 协议承载； m=2：使用 TLS 协议承载。 n：设置 SIP 承载的网络侦听端口，默认为 5060，取值范围为：1024 ~ 65536。 t_L：RTP 端口范围下限，取值范围：1024≤t_L，缺省值为 6000。 t_H：RTP 端口范围上限，取值范围：t_H≤65535，缺省值为 10000。 p_1,p_2,…. p_n：通话使用的语音编解码列表，缺省值为 6,7,131，其中 n≤10，值在前表示优先级高。其值的意义如下： p_x=6：G.711 μ-Law 格式； p_x=7：G.711 A-Law 格式； p_x=131：G.729A 格式。
功能描述	设置 SIP 信令的工作参数
注意事项	只有当配置项ProtocolType设置为 1 时，本配置项才有效。

3.1.2.28.1.2 设置注册服务器

配置项	**Register** **UserName** **RegPassword** **Domain** **RegExpires**
节	**[SIP]**
书写格式	Register=m UserName=s RegPassword=j Domain=x RegExpires=q
取值范围	m：表示是否启动注册功能。 m=0：不启动注册功能； m<>0：启动注册功能。 s：用户名，最大长度为 50 的字符串。 当 m<>0 时，s 表示注册用户名；当 m=0 时，s 表示 SIP URI 中 username 字段的内容。 j：注册用户密码。 当 m<>0 时有效，表示注册用户密码。 x：注册服务器。 当 m<>0 时有效，表示注册服务器的地址。格式为 host[:port] q：注册有效期。表示板卡每隔多长时间向注册服务器刷新注册消息。
功能描述	设置注册服务器
注意事项	只有当配置项ProtocolType设置为 1 时，本配置项才有效。

3.1.2.28.1.3 设置其他参数

配置项	**LogLevel** **LogFile** **EventThreadNum** **UdpHeartTime**
节	**[SIP]**
书写格式	LogLevel=m LogFile=s EventThreadNum=j UdpHeartTime=x

取值范围	m：设置输出的日志等级，具体为： m=1：SEGMENT，严重错误等级，默认值； m=2：FATAL，协议栈消息级错误； m=3：ERROR，逻辑错误； m=4：WARNING，逻辑警告； m=5：INFO1，逻辑信息； m=6：INFO2，SIP 消息信息； m=7：INFO3，RTP 消息信息。 s：设置输出的日志文件，长度小于 256 个字符。 j：设置捕获 SIP 协议事件的线程池大小，默认值为 3，范围为：0<j<6。 x：设置 UDP 端口保持心跳的时间。取值范围为：0≤x≤1000，单位为 ms。x=0：不使用心跳功能（默认值）。
功能描述	设置其他参数
注意事项	只有当配置项ProtocolType设置为 1 时，本配置项才有效。

3.1.2.28.2H.323 信令的高级设置

3.1.2.28.2.1 设置 H.323 信令的工作参数

配置项	**LocalPort** **RTPRange** **AudioCodecList** **FastStart** **Tunnel**
节	**[H.323]**
书写格式	LocalPort=n RTPRange= t_L, t_H AudioCodecList= p_1,p_2,…. p_n FastStart=j Tunnel=h
取值范围	m：设置 H.323 信令承载的网络协议，具体为： m=0：使用 UDP 协议承载； m=1：使用 TCP 协议承载，默认值； m=2：使用 TLS 协议承载。 n：设置 H.323 承载的网络侦听端口，默认为 1720，取值范围为：1024 ~ 65536。 t_L：RTP 端口范围下限，取值范围：1024≤t_L，缺省值为 5000。 t_H：RTP 端口范围上限，取值范围：t_H≤65535，缺省值为 5999。 p_1,p_2,…. p_n：通话使用的语音编解码列表，缺省值为 6,7,131，其中 n≤3，值在前表示优先级高。其值的意义如下： p_x=6：G.711 μ-Law 格式； p_x=7：G.711 A-Law 格式； p_x=131：G.729A 格式。 j：设置 H.323 通话快速启动标志，具体为： j=0：不使用快速启动； j=1：使用快速启动，默认值。 h：设置 H.323 隧道使能标志，具体为： h=0：不使用 H.323 隧道使能标志； h=1：使用 H.323 隧道使能标志，默认值。
功能描述	设置 H.323 信令的工作参数
注意事项	只有当配置项ProtocolType设置为 0 时，本配置项才有效。

3.1.2.28.2.2 设置 H.323 服务器

配置项	**GateKeeper** **GKPassword**
节	**[H.323]**
书写格式	GateKeeper=m GKPassword=n
取值范围	m：设置 H.323 GK，长度小于 256 个字符；具体为： m=disable：不使用 GK，默认值； m=discover：在本网络内自动寻找 GK； m=[ipv4/hostname]：指定 GK。 n：设置本端向 GK 注册时的密码，长度小于 16 个字符。
功能描述	设置 H.323 服务器
注意事项	只有当配置项ProtocolType设置为 0 时，本配置项才有效。

3.1.2.28.2.3 设置其他参数

配置项	**JitterDelay**
节	**[H.323]**
书写格式	JitterDelay=m,n
取值范围	m：设置 Jitter Buffer 抖动范围的下限，取值范围：50≤m≤100，缺省值为 50。 n：设置 Jitter Buffer 抖动范围的上限，取值范围：250≤n≤300，缺省值为 250。
功能描述	设置 H.323 其他参数
注意事项	只有当配置项ProtocolType设置为 0 时，本配置项才有效。

3.1.2.29 DST 系列板卡常用配置项(REC 系列)

配置项	**SupplyBoardClockLine**
节	**[BoardId=x]**
书写格式	SupplyBoardClockLine=n
取值范围	n：板卡通道号，从 0 开始编号。
功能描述	设置板卡上提供参考时钟的电话线。若该配置项不配置或者配置值不是合法值，板卡则自动选择参考时钟。
注意事项	● 本配置项是高级配置项，只适用于 DST 系列板卡。

3.1.2.30 ATP 系列板卡常用配置项(REC 系列)

3.1.2.30.1 设置输入端的增益

配置项	**MicGain**
节	**[BoardId=x]**
书写格式	格式 1：MicGain=g,g,g,g 格式 2：MicGain=g,g,g,g,g,g,g,g 格式 3：MicGain=g,g,g,g,g,g,g,g,g,g,g,g,g,g,g,g
取值范围	g=0：正常增益； g=1：高增益。
功能描述	设置模拟录音通道上输入信号的增益。如果语音信号来自模拟电话线路，应将输入增益设置为 0 正常增益(0 DB)；如果来自动圈式话筒，使用高增益(20DB)。
注意事项	● 格式 1 适用于 ATP 系列的 USB 录音盒； ● 格式 2 适用于 8 通道规格的 ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 ATP 系列板卡。

3.1.2.30.2 设置空闲通道的能量检测

配置项	**EnableIdleChTA**
节	**[BoardId=x]**
书写格式	格式 1：EnableIdleChTA =b,b,b,b 格式 2：EnableIdleChTA =b,b,b,b,b,b,b,b 格式 3：EnableIdleChTA =b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0：关闭（缺省值）； b=1：开启。
功能描述	设置模拟录音通道在空闲时是否驱动能量并接收 DTMF 码。
注意事项	● 格式 1 适用于 ATP 系列的 USB 录音盒； ● 格式 2 适用于 8 通道规格的 ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 ATP 系列板卡。

3.1.2.30.3 设置预录音功能

配置项	**PrerecordEnable**
节	**[SystemConfig]**
书写格式	PrerecordEnable=b
取值范围	b=0：关闭（缺省值）； b=1：开启。
功能描述	开启或关闭预录音功能。
注意事项	● 本配置项只适用于 ATP 系列板卡； ● 要求 SynCTI Ver. 2.1 或更高。

配置项	**PrerecordMode**
节	**[SystemConfig]**
书写格式	PrerecordMode=m
取值范围	m=0：当 Barge in 检测器检测到 Barge in 时开始将录音数据写入到内部缓冲区； m=1：当模拟录音通道在线路上检测到摘机动作时开始将录音数据写入到内部缓冲区。 缺省值为 0。
功能描述	设置预录音功能的工作模式。
注意事项	● 本配置项只适用于 ATP 系列板卡； ● 要求 SynCTI Ver. 2.1 或更高。

配置项	**PrerecordInsertTime**
节	**[SystemConfig]**
书写格式	PrerecordInsertTime=t
取值范围	0<t<1000，单位为毫秒，缺省值为 500。
功能描述	预录音功能启动后，当应用程序启动文件录音任务时，本参数设置将预录音的数据写入到文件中的时间长度。
注意事项	● 本配置项只适用于 ATP 系列板卡； ● 要求 SynCTI Ver. 2.1 或更高。

配置项	**PrerecordCodec**
节	**[SystemConfig]**
书写格式	PrerecordCodec=m
取值范围	m=6： A-law (缺省值)； m=7： μ-law； m=17：IMA ADPCM； m=49：GSM 6.10。
功能描述	设置预录音的编码格式。
注意事项	● 本配置项只适用于 ATP 系列板卡； ● 要求 SynCTI Ver. 2.1 或更高。

3.1.2.30.4检测脉冲方式的按键信号

配置项	**EnablePulseKeyDetect**
节	**[SystemConfig]**
书写格式	EnablePulseKeyDetect=b
取值范围	b=0：否（缺省值）； b=1：是。
功能描述	设置是否需要检测脉冲方式的按键信号。
注意事项	本配置项只适用于模拟中继线录音通道。

3.1.2.30.5设置闪断信号的最大保持时间

配置项	**MaxRecChFlashFilterTime**
节	**[SystemConfig]**
书写格式	MaxRecChFlashFilterTime=t
取值范围	t：保持时间，单位：毫秒，缺省值为 200 毫秒。
功能描述	设置闪断信号的最大保持时间。更多信息请参见第 1 章中“模拟电话线的电压变化”部分内容。

3.1.2.30.6设置振铃信号计数器的清零保护时间

配置项	**RecChClearRingDelayTime**
节	**[SystemConfig]**
书写格式	RecChClearRingDelayTime=t
取值范围	t≥0，单位为秒，缺省值为 0。
功能描述	设置振铃信号计数器的清零保护时间。

3.1.2.30.7设置软交换功能

配置项	**BusPlayListen**
节	**[SystemConfig]**
书写格式	BusPlayListen=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	驱动程序是否开启软交换功能。更多信息请参见第 1 章中“ATP系列板卡的工作原理框图”部分内容。

3.1.2.31 DTP 系列板卡常用配置项(REC 系列)

3.1.2.31.1将 ISDN 消息保存到文件

配置项	**SpyPrintLog**
节	**[SpyPcm]**
书写格式	SpyPrintLog=b
取值范围	b=0：否(缺省值)； b=1：是。
功能描述	设置是否将驱动程序收到的 1 帧消息输出到日志文件中。如果设置为是，每当驱动程序收到 1 帧 ISDN 消息，都会将其写入到应用程序当前目录下的 SpyIFrame.log 文件中。
注意事项	每当日志文件中的消息总数超过 100,000 时，驱动程序会自动将日志文件中的数据删除，重新开始记录消息。

3.1.2.31.2SS7 信令状态机同步输出原始信令消息

配置项	**bOpenSpySS7Msu**
节	**[SS7Spy]**
书写格式	bOpenSpySS7Msu=b

取值范围	b=0：否（缺省值）； b=1：是。
功能描述	SynCTI驱动程序的ISUP/TUP协议呼叫状态机在处理收到的ISUP/TUP消息时，是否同时将原始消息输出给应用程序。更多信息请参见第 1 章中“DTP系列板卡的工作原理框图”部分内容。
注意事项	● 本配置项只适用于 DTP 系列板卡的 SS7 信令； ● 本配置项要求 SynCTI 驱动程序的版本为 4.7.2.0 或更高。

3.1.3 高级配置项

3.1.3.1 驱动程序支持多个进程

配置项	**MultiCardMultiProcess**
节	**[SystemConfig]**
书写格式	MultiCardMultiProcess=b
取值范围	b=0：否(缺省值)； b=1：是。
功能描述	设置 SynCTI 驱动程序是否支持多进程方式的应用程序

3.1.3.2 输出 API 函数调用的调试信息

配置项	**EnableDebugAPI**
节	**[DebugView]**
书写格式	EnableDebugAPI=b
取值范围	b<1 或者 b>4（缺省值），不输出调试信息； b = 1，输出 API 函数调用信息到 DebugView； b = 2，输出 API 函数调用信息到 LOG 文件； b = 3，输出事件信息到 LOG 文件； b = 4，同时输出 API 函数调用和事件信息到 LOG 文件。
功能描述	设置是否输出应用程序调用 API 函数的调试信息。如果允许驱动程序输出 API 函数的调用信息，驱动程序会将应用程序发出的所有 API 函数调用的函数名称、入口参数、函数执行结果等信息以字符串的形式进行输出。 更多信息请参见第 1 章中“使用驱动程序提供的调试功能”部分内容。
注意事项	● 对于命令型的函数调用，驱动程序会输出每一次函数信息；对于查询类型的函数调用，只有当查询操作的返回值发生变化后，才会输出； ● 开启功能会消耗主机上 CPU 资源，因此通常只用于调试应用程序。当应用系统正式运行时，请关闭此项功能。

配置项	**SetEventRange**
节	**[DebugView]**
书写格式	SetEventRange=n..m
取值范围	n：表示写 LOG 的起始事件号，n=-1(缺省值)表示输出<m 的事件； m：表示写 LOG 的结束事件号,m=-1(缺省值)表示输出>n 的事件。
功能描述	在设置 EnableDebugAPI 的条件下，设置事件输出写 LOG 的事件范围，从 n 到 m 之间的事件输出都写到 Log 中（含 n 和 m），如果 n 和 m 都是-1,表示所有事件输出都写 LOG。
注意事项	只有当配置项EnableDebugAPI设置为输出时，本配置项才有效。

配置项	**SetChRange**
节	**[DebugView]**
书写格式	SetChRange =n..m
取值范围	n：表示事件输出写 LOG 的起始通道号，n=-1(缺省值)表示输出通道范围为<m； m：表示事件输出写 LOG 的结束通道号，m=-1(缺省值)表示输出通道范围为>n。
功能描述	在设置 EnableDebugAPI 的条件下，设置事件输出写 LOG 的通道范围，从 n 到 m 之间的通道具备事件输出写 Log 能力（含 n 和 m），如果 n 和 m 都是-1,表示所有通道都具备事件输出到 LOG 能力。
注意事项	只有当配置项EnableDebugAPI设置为输出时，本配置项才有效。

配置项	**Mask_SsmGetNoSoundTime** **Mask_SsmGetChStateKeepTime** **Mask_SsmGetPlayedTime** **Mask_SsmGetPlayedPercentage** **Mask_SsmGetPlayOffset** **Mask_SsmGetRecTime** **Mask_SsmGetRecOffset**
节	**[DebugView]**
书写格式	Mask_SsmGetNoSoundTime=b Mask_SsmGetChStateKeepTime=b Mask_SsmGetPlayedTime=b Mask_SsmGetPlayedPercebtage=b Mask_SsmGetPlayOffset=b Mask_SsmGetRecTime=b Mask_SsmGetRecOffset=b
取值范围	b=0：不输出（缺省值）； b=1：输出。
功能描述	设置驱动程序是否输出一些查询类型的 API 函数的调用信息。在配置项的名称中，“Mask_”后面的字符串就表示函数名称。 更多信息请参见第 1 章中“使用驱动程序提供的调试功能”部分内容。
注意事项	只有当配置项EnableDebugAPI设置为输出时，本配置项才有效。

3.2 预加载语音配置文件 ShIndex.ini（CTI 系列）

3.2.1 [System]节

配置项	**MaxIndexSeg**
节	**[System]**
书写格式	MaxIndexSeg=n
取值范围	0≤n≤65535，缺省值为 0。
功能描述	设置预加载语音片段的总数。
注意事项	MaxIndexSeg 决定了[SegNo=x]节的个数。

3.2.2 [SegNo=x]节

设置每段内存索引语音片段的文件信息，其中 x 为段号。如果有多个语音片段，需要配置多个[SegNo=x]节。

配置项	**FileName**
节	**[SegNo=x]**
书写格式	FileName=s
取值范围	s为包含语音片段的文件名，可以是标准的wav格式文件，也可以是没有文件头的无格式文件。如果是标准的wav文件，配置项CodecFormat指定的格式将被忽略；否则被视为无格式文件，编码格式由配置项CodecFormat指定。 如果不指定全路径，驱动程序在应用程序的当前目录下查找该文件。
功能描述	设置语音文件名。
使用范例	FileName=d0.wav

配置项	**Alias**
节	**[SegNo=x]**
书写格式	Alias=s

取值范围	s 为别名，长度不能超过 20 个字符，不能以数字‘0’～‘9’作为别名字符串的首字符。为空表示不设置别名。 缺省值为空。
功能描述	为语音片段分配别名。
使用范例	Alias=D0

配置项	**CodecFormat**
节	**[SegNo=x]**
书写格式	CodecFormat=n
取值范围	n=6： A-Law（缺省值）； n=7： U-Law； n=17：IMA ADPCM。
功能描述	设置语音片段的语音编码格式。
注意事项	所有的预加载语音片段都必须具有相同的语音编码格式。

配置项	**StartOffset**
节	**[SegNo=x]**
书写格式	StartOffset=n
取值范围	n：语音片段数据在文件中的起始位置偏移量，缺省值为 0，对于 ADPCM 格式应该设置为 256 的整数倍。
功能描述	设置从文件中实际的语音数据部分开始计算的语音片段的起始位置。对于标准的 wav 文件，起始位置为 0 是指文件头后的第 1 个语音数据字节；对于无格式文件，是从文件的头部开始计算。

配置项	**Length**
节	**[SegNo=x]**
书写格式	Length=n
取值范围	N：语音片段的长度，单位为字节，缺省值为-1。
功能描述	设置本语音片段的数据长度。语音片段的字节长度与时间长度之间的换算值取决于编码格式，相关内容请参见第 1 章中的“Synway板卡支持的语音编码/解码格式”部分。
注意事项	● 如果本参数大于配置项StartOffset到文件尾的数据长度，则实际加载的语音片段长度为StartOffset； ● 如果 Length 设置为-1，表示从 IStartPos 装载到文件尾。

4 SS7 信令方式（CTI 系列）

4.1 SS7 信令方式的基本概念

4.1.1 OPC 和 DPC

信令点(Signaling Point, SP)是信令网中处理控制消息的节点。产生消息的信令点称为该消息的起源点 OPC（Originating Point Code），在配置项 OPC 中进行设置；消息到达的信令点称为该消息的目的点 DPC（Destination Point Code）。OPC、DPC 均在 SS7 信令服务器的 Ss7Server.ini 文件中设置。

信令点的编码方式包括中国国内网的信令点编码和国际信令网的信令点编码。对于国际信令点编码，CCITT 建议采用 14bit 编码，编码方式见下图：

3bit	8bit	3bit
大区或洲 （ZONE）	国家或地区 （AREA）	信令点 （POINT）

对于中国的信令点编码采用 24bit 方式，各比特的含义如下图所示：

8bit	8bit	8bit
主信令区编码	分信令区编码	信令点编码

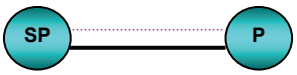
Synway的SHD系列板卡同时支持国际信令点编码方式和中国信令点编码方式，可通过配置项SpCodeLen中进行设置。

4.1.2 直连和准直连

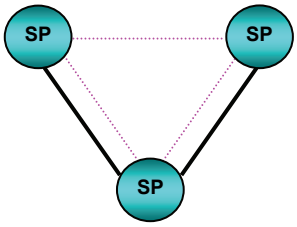
两个信令点之间的信令消息，通过直接连接两个信令点的信令链路进行传递，称为直联工作方式。

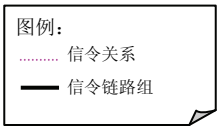
属于某信令关系的消息，经过两个或多个串接的信令链路传送，中间要经过一个或几个信令转接点，但通过信令网的消息所取的通路在一定时间内是预先确定的和固定的，称为准直联。

下图表示了直联和准直联工作方式的概念：



(a)直连工作方式





(a)准直连工作方式

SynCTI驱动程序支持直连和准直连的工作方式，可通过配置配置项MaxDPC/DPC及MaxUP_DPC/UP_DPC进行设置。更多信息请参见本章中“SS7 信令服务器的配置”部分内容。

4.1.3 信令链路和链路组

在两个信令点之间传送信令消息的链路称为信令链路。直接连接两个信令点的一束信令链路构成一个信令链路

组。

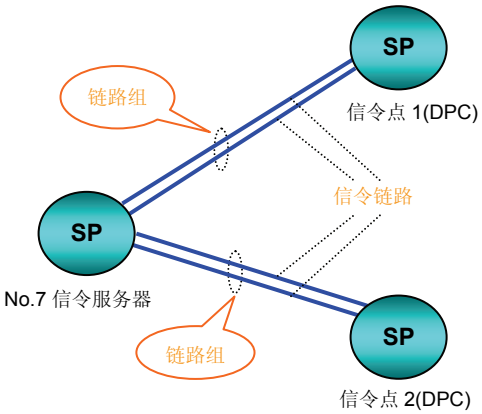
SynCTI驱动程序最多 48 个信令链路组，可通过配置项MaxLinkSet和LinkSet进行设置；每个信令链路组最多支持 2 条 64kbps信令链路，可通过配置项MaxSs7Pcm和Ss7PcmLink进行设置。如果信令链路组包含 2 条信令链路，这 2 条信令信令链路均工作于负荷分担方式。

SS7 信令服务器最多可以连接 48 个信令点或信令转接点。

4.2 Synway 板卡的 SS7 信令应用系统

SynCTI 驱动程序以 SS7 信令服务器的方式向应用程序提供 SS7 信令的服务，应用程序作为客户端程序，可以有一个或多个，可以位于一台或多台计算机上，甚至可以与服务器运行于同一台计算机上。

SS7 信令服务器为使用中国 SS7 信令的数字中继电路提供信令服务，支持本地信令点到多个信令点的连接。有关信令链路、链路组、信令点编码 DPC/OPC、信令点路由的关系如下图所示：



本 SS7 信令服务器程序具有下列特点：

- 一条数字中继线可以同时传送信令消息和语音信息。通常使用数字中继线的 16 时隙来传送信令消息，时隙 1~15、17~31 这 30 个时隙用来传送语音信息；
- 每个链路组最多可以包括 2 条 64kbps 信令链路，这 2 条信令链路之间采用负荷分担方式，由服务器自动进行负荷管理；
- 本机信令点最多可以同时连接到 48 个信令点（SP 或 STP），并且本机信令点与其它直联信令点之间只能使用 1 个链路组（即 1 条路由）；
- 采用 C/S 架构，通过 TCP/IP 协议支持分布式信令链路/客户端结构，但单机使用（服务器与板卡信令链路在同一台计算机上）时可以不需要网卡；
- 如果 SS7 信令服务器上包含了物理信令链路，服务器程序在启动时，会自动为 SHD 系列板卡加载 SynCTI 驱动程序。**注意：在这种情况下，SHD 系列板卡上数字中继线的话路时隙无法再作为话路使用。**

SS7 信令服务器包括下列文件：

Ss7Server.ini	SS7 信令服务器的配置文件
Ss7Monitor.exe	SS7 信令服务器主程序
Ss7Server.dll	SS7 信令调度模块
Mtp3.dll	SS7 信令 MTP3 层模块

TcpServer.dll	SS7 信令服务器与客户端通信模块 1
MmfServer.dll	SS7 信令服务器与客户端通信模块 2

4.3 SS7 信令服务器的配置

SS7 信令服务器的执行程序为 Ss7Monitor.exe，它使用的配置文件为 Ss7server.ini。

4.3.1 设置信令服务器的信令点编码

配置项	OPC
节	[Ss7SystemConfig]
书写格式	OPC=a.b.c
取值范围	a：主信号区编码，以十进制表示； b：分信号区编码，以十进制表示； c：信号点编码，以十进制表示；
功能描述	设置信令服务器的信令点编码，通常由电信局分配
注意事项	本配置项为必须配置项
使用范例	OPC=1.2.13

配置项	SpCodeLen
节	[Ss7SystemConfig]
书写格式	SpCodeLen=n
取值范围	n：信令点的编码方式， n=24（缺省值）：使用中国的信令点编码方式； n=14：使用国际的信令点编码方式
功能描述	设置信令点的编码方式
注意事项	中国的用户通常无须设置本配置项

4.3.2 设置信令服务器的 IP 地址

配置项	ServerIP SecondServerIP
节	[Ss7SystemConfig]
书写格式	ServerIP=a.b.c.d SecondServerIP=a.b.c.d。
取值范围	a.b.c.d：IP 地址。ServerIP 的缺省值为 127.0.0.1。
功能描述	设置 SS7 信令服务器的 IP 地址。ServerIP 用于设置信令服务器本机 IP 地址，SecondServerIP 用于设置与本机互为热备的信令服务器机器 IP 地址。
注意事项	如果只有一个信令服务器，无须设置 SecondServerIP。如果设置了 SecondServerIP，本信令服务器将以热备份服务器的方式启动。

配置项	Port
节	[Ss7SystemConfig]
书写格式	Port=n
取值范围	n：端口号，缺省值为 5150。
功能描述	设置 SS7 信令服务器的侦听端口号。
注意事项	如果只有一个信令服务器，无须设置本配置项。

4.3.3 设置 MTP3 的工作模式和工作参数

配置项	ConfigMtp3AsSTP
节	[Ss7SystemConfig]
书写格式	ConfigMtp3AsSTP=m

取值范围	m=0：信令点（缺省值），Mtp3 层对接收消息的 DPC 与本机该链路组的 OPC 作一致性检查。 m=1：信令转接点，Mtp3 层对接收消息的 DPC 不作检查。
功能描述	设置信令服务器中 MTP3 的工作模式。

配置项	SendSNT
节	[Ss7SystemConfig]
书写格式	SendSNT=n
取值范围	n=0：不发送（缺省值） n=1：发送
功能描述	设置 MTP3 的工作参数：是否定期向对端交换机性发送信令链路测试消息(SLTM)

配置项	SubServicefield
节	[Ss7SystemConfig]
书写格式	SubServicefield=0xn
取值范围	n：十六进制形式的子业务编码，低四 bit 有效（Bit3~Bit0 分别记为 DCBA）： DC：网络表示语 00：国际网路 01：国际备用 10：国内网路 11：国内备用 BA：备用。 n 的缺省值为 8。
功能描述	设置 SS7 子业务编码
使用范例	SubServicefield=0x8

4.3.4 设置客户端的信息

配置项	MaxSs7Client IP
节	[Ss7ClientInfo]
书写格式	MaxSs7Client=M IP[m]=a.b.c.d
取值范围	M：　使用 SS7 信令服务器的客户端计算机的总数，缺省值为 0。如果 M 大于 0， IP 配置项必须出现 M 次； m：　客户端的逻辑编号，从 0 开始编号，0≤m<M； a.b.c.d：第 m 个客户端计算机的 IP 地址。
功能描述	MaxSs7Client 设置使用 SS7 信令的客户端计算机的总数，IP 逐项设置每个客户端计算机的 IP 地址。单机应用可以设置本配置项为 1
使用范例	如果 SS7 信令服务器与客户端程序运行在同一台计算机上： **[Ss7ClientInfo]** MaxSs7Client=1 IP[0]=127.0.0.1 如果有 2 台客户端计算机连接到 SS7 信令服务器： **[Ss7ClientInfo]** MaxSs7Client=2 IP[0]=203.203.203.200 IP[1]= 203.203.203.201

4.3.5 设置信令链路的物理位置

配置项	**MaxSs7Pcm** **Ss7PcmLink**
节	**[Ss7PcmLinkInfo]**
书写格式	MaxSs7Pcm=M Ss7PcmLink[m]=IP[n],LocalPCM[k]
取值范围	M：信令链路总数。SS7 信令服务器最多支持 48 个链路组，每个链路组最多支持 2 条信令链路，因此 M 的取值范围为 0≤M≤96； m：信令链路在服务器中的逻辑编号（从 0 开始编号），0≤m<M； n：包含第m条信令链路的客户端计算机的逻辑编号，该逻辑编号在配置项IP中定义； k：服务器上第m条信令链路在该链路所在客户端计算机上的逻辑编号。k可以通过系统配置文件ShConfig.ini中的配置项TotalPcm和Pcm进行设置；
功能描述	MaxSs7Pcm 设置 64kbps 信令链路的总数，Ss7PcmLink 设置 SS7 信令服务器中信令链路在客户端计算机中的物理位置
使用范例	假设应用系统中包含 1 台 SS7 信令服务器和 2 台客户端计算机，总共有 2 条信令链路，第 1 条信令链路的物理位置为第 1 台客户端计算机，其本地编号为 0；第 2 条信令链路的物理位置为第 2 台客户端计算机，其本地编号为 3；则相关配置项的设置如下： **[Ss7ClientInfo]** MaxSs7Client=2 IP[0]=203.203.203.200 IP[1]= 203.203.203.201 **[Ss7PcmLinkInfo]** MaxSs7Pcm=2 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[1],LocalPCM[3] 如果 2 条信令链路均位于第 1 台客户端计算机，并且它们的本地编号分别为 0 和 1，则 MaxSs7Pcm 和 Ss7PcmLink 的设定值为： Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[1]

4.3.6 设置信令链路组

配置项	**MaxLinkSet** **LinkSet**
节	**[LinkSetInfo]**
书写格式	MaxLinkSet=N 格式 1：LinkSet[n]=Ss7PcmLink[i] 格式 2：LinkSet[n]=Ss7PcmLink[i]+Ss7PcmLink[j] 格式 3：LinkSet[n]=Ss7PcmLink[i],a.b.c（链路组中只有 1 条信令链路） 格式 4：LinkSet[n]=Ss7PcmLink[i]+Ss7PcmLink[j],a.b.c（链路组中有 2 条信令链路） **注：格式 1 和格式 2 分别适用于链路组中只有 1 条和 2 条信令链路的情况；**
取值范围	N： 服务器上信令链路组的总数，1≤N≤48，N 决定了 LinkSet 配置项的数目； n： 链路组的逻辑序号，从 0 开始编号，0≤n<N； i,j： 第n个信令链路组中包含的信令链路在服务器上的逻辑编号，在配置项Ss7PcmLink中指定； a.b.c：为第 n 个信令链路组分配的 OPC。
功能描述	MaxLinkSet 设置连接到本服务器的信令链路组的总数，LinkSet 设置链路组由哪些信令链路组成。
注意事项	● 格式 1 和格式 3 适用于链路组中只有 1 条信令链路的情况，格式 2 和格式 4 适用于链路组中有 2 条信令链路的情况； ● 链路组的OPC由配置项LinkSet[n]优先指定。如果配置项LinkSet中未指定OPC，则该链路组的OPC由配置项OPC指定。

使用范例	MaxLinkSet=2 LinkSet[0]=Ss7PcmLink[1],1.1.1 LinkSet[1]=Ss7PcmLink[2]+ Ss7PcmLink[0],1.1.2

4.3.7 设置 DPC

配置项	**MaxDPC** **DPC**
节	**[DPCInfo]**
书写格式	MaxDPC=N DPC[n]=x.y.z,LinkSet[k]
取值范围	N： DPC 的总数，0≤N≤48； n： DPC 的逻辑编号，从 0 开始编号，0≤n<N； x.y.z：第 n 个 DPC 的信令点编码（十进制），z 为主信号区编码，y 为分信号区编码，z 为信号点编码，由电信局分配。 k： 连接本信令服务器与第n个DPC的信令链路组的逻辑编号。信令链路组的逻辑编号在配置项MaxLinkSet和LinkSet中指定。
功能描述	配置项 MaxDPC 设置连接到本服务器的 DPC 总数(SP 或 STP)；配置项 DPC 设置 DPC 的编码，以及到达该 DPC 的路由信息。
注意事项	● 如果 N≥1，DPC 必须从 n=0 开始连续配置 N 次； ● 本配置项只适用于直连方式。准直连方式请使用配置项MaxUP_DPC和UP_DPC。
使用范例	MaxDPC=2 DPC[0]=1.1.8,LinkSet[0] DPC[1]=1.1.9,LinkSet[1]

4.3.8 设置用户层消息的目的点码

配置项	**MaxUP_DPC** **UP_DPC**
节	**[UP_DPCInfo]**
书写格式	MaxUP_DPC=N 格式 1：UP_DPC[n]=x.y.z,LinkSet[i] 格式 2：UP_DPC[n]= x.y.z,LinkSet[i]+LinkSet[j] 格式 3：UP_DPC[n]= x.y.z,LinkSet[i]+LinkSet[j],m
取值范围	N： 配置项 UP_DPC 的个数，0≤N≤48； n： UP_DPC 的逻辑编号，从 0 开始编号，0≤n<N； x.y.z：第 n 个 UP_DPC 的信令点编码（十进制），z 为主信号区编码，y 为分信号区编码 z 为信号点编码，由电信局分配； i,j： 连接本信令服务器与第n个UP_DPC之间的信令链路组的逻辑编号。信令链路组的逻辑编号在配置项MaxLinkSet和LinkSet中指定； m: 如果连接本信令服务器与第 n 个 UP_DPC 之间的信令链路组有 2 组，m 用于指定服务器使用信令链路组 i 和 j 的工作方式： m=0：负荷分担（缺省值，此时格式 2 与格式 3 等效）； m=1：第 i 组信令链路组为正常路由，第 j 路由为迂回路由。
功能描述	MaxUP_DPC 设置通过信令转接点（STP）与本系统相联的信令业务点（SP）总数，它的设定值决定了配置项 UP_DPC[n]出现的行数。UP_DPC 设置第 n 个信令点的编码，以及到达该信令点的路由信息。
注意事项	● 如果 N≥1，UP_DPC 必须从 n=0 开始连续配置 N 次； ● 本配置项可以用于直连、准直连、直连/准直连混合环境； ● 如果将本配置应用于直联信令点，必须与配置项MaxDPC、DPC的配置参数保持一致，即 UP_DPC 与相应的DPC应指向同一SP。 ● 设置 UP_DPC 的格式取决于本信令服务器与第 n 个 UP_DPC 之间的信令链路组的数量： ✧ 只有 1 组：使用格式 1 ✧ 有 2 组：使用格式 2 或格式 3。

使用范例	MaxUP_DPC=2 UP_DPC[0]=1.2.8,LinkSet[0] UP_DPC[1]=1.2.9,LinkSet[1]

4.3.9 设置 TUP/ISUP 消息的分发路由

配置项	**UP_DPC,CIC_PCM** **DPC,CIC_PCM** **DefaultTupRouterClientIpId** **DefaultIsupRouterClientIpId**
节	**[TUPRouter] / [ISUPRouter]**

书写格式	格式 1：UP_DPC[i],CIC_PCM[p]=IP[j],LocalPCM[k] //适用于 TUP/ISUP 协议
	格式 2：DPC[i],CIC_PCM[p]=IP[j],LocalPCM[k] //适用于 TUP/ISUP 协议
	DefaultTupRouterClientIpId=c //适用于 TUP 协议
	DefaultIsupRouterClientIpId=c //适用于 ISUP 协议

取值范围	i： DPC或UP_DPC的逻辑编号。格式 1 指定UP_DPC，此时配置项MaxUP_DPC必须大于 0，应用于直联或准直联环境；格式 2 指定DPC，此时可以不设置配置项MaxUP_DPC或将其设置为 0，只应用于直联环境。 p： CIC 字段中的数字中继线编号，由电信局分配的电路 CIC 起始编号除以 32 后得到的值。不必从 0 开始编号。 j,c：客户端计算机的IP地址的逻辑编号。更多信息请参见配置项MaxSs7Client和IP的说明。c的缺省值为 0。 k： 数字中继线在客户端计算机上的逻辑编号。更多信息请参见配置项TotalPcm和Pcm的说明。
功能描述	UP_DPC,CIC_PCM 设置 TUP/ISUP 消息的派发路由。 SS7 信令服务器分发 TUP/ISUP 消息的过程为： ✧ 每当 SS7 信令服务器从第 i 个 UP_DPC / DPC 收到一条 TUP/ISUP 消息时，服务器会从 CIC 字段中获取数字中继线的编号 p。如果能够为此 TUP/ISUP 消息找到设定的路由，服务器会按照配置项 UP_DPC,CIC_PCM 设定的路由，将此 TUP/ISUP 消息的 CIC 字段中的数字中继线编号 p 修改为 k，并将此 TUP 消息转发给客户端 j；如果不能找到，服务器会进一步检查配置项 DefaultTupRouterClientIpId(对于 TUP 协议)或 DefaultIsupRouterClientIpId（对于 ISUP 协议）的设置：如果已经设置，此 TUP/ISUP 消息将被转发给客户端 c，否则，此 TUP/ISUP 消息将被丢弃。 ✧ 每当 SS7 信令服务器从客户端 j 收到一条 TUP/ISUP 消息时，服务器会从 CIC 字段中获取数字中继线的编号 k，并按照配置项 UP_DPC,CIC_PCM 设定的路由，将此 TUP/ISUP 消息的 CIC 字段中的数字中继线编号 k 修改为 p，然后将此 TUP/ISUP 消息转发给第 i 个 UP_DPC / DPC。如果不能找到匹配的路由，此 TUP/ISUP 消息将被丢弃。
注意事项	● TUP 协议使用**[TUPRouter]**配置节，ISUP 协议使用**[ISUPRouter]**配置节； ● 如果没有设置[UP_DPCInfo]节，或者 MaxUP_DPC=0，则应使用 DPC,CIC_PCM：DPC[i],CIC_PCM[p]=IP[j],LocalPCM[k]； ● 客户端的应用程序如果使用驱动程序提供的 TUP/ISUP 协议，必须设置本配置项； ● 必须用本配置项对电信局分配的每条数字中继线逐个进行设置。
使用范例	[ISUPRouter] DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0] DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[1]

4.3.10 设置服务器的启动特性

配置项	**ConfigAsGateway**
节	**[Monitor]**
书写格式	ConfigAsGateway=n
取值范围	n=0：否（缺省值） n=1：是

功能描述	当 SS7 信令服务器启动时，是否加载 SynCTI 驱动程序。如果服务器中没有安装任何 Synway 板卡，应当将本配置项设置为 0；如果安装了提供信令链路的 SHD 系列板卡，必须将此配置项设置为 1，当服务器启动时，服务器会自动加载 SynCTI 驱动程序。 在加载驱动程序时，SS7 信令服务器使用的配置文件名为 ShConfig.ini 和 ShIndex.ini。服务器首先在当前目录下查找这两个配置文件。如果找不到，则到 SynCTI 驱动程序安装目录下查找。如果还是找不到，服务器启动失败，会返回出错信息。有关 ShConfig.ini 和 ShIndex.ini 的配置方法请参考第 4 章。
注意事项	如果本配置项设置为 1，则在服务器中运行的其它应用程序（如果有）不能加载 SynCTI 驱动程序。

4.3.11 设置服务器的显示特性

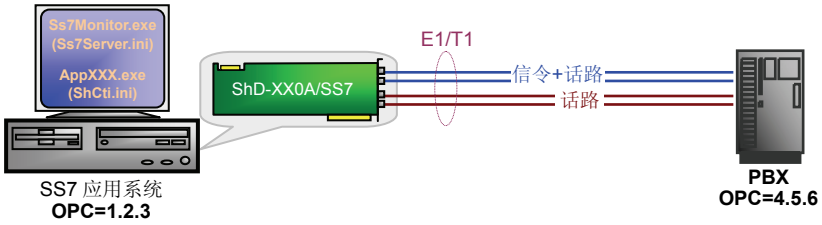
配置项	**RcvMsuListMaxItem TxMsuListMaxItem**
节	**[Monitor]**
书写格式	RcvMsuListMaxItem=n　　//适用于用户部分（客户端）到服务器的消息 TxMsuListMaxItem=m　　//适用于信令点（SP/STP）到服务器的消息
取值范围	n,m：允许显示的最大消息数量，缺省值为 50 条
功能描述	SS7 信令服务器可以将来自信令点（SP/STP）或用户部分（客户端）的消息显示在屏幕上，本配置项用于设置可显示消息的最大值。 TxMsuListMaxItem 用于设置用户部分（客户端）到服务器的消息，RcvMsuListMaxItem 用于设置信令点（SP/STP）到服务器的消息。

4.4 SS7 信令服务器的应用及配置实例

4.4.1 单 OPC/单 DPC

4.4.1.1 单机单卡的应用系统

在一台计算机上安装 1 片 SHD 系列板卡的最小应用系统，如下图所示：

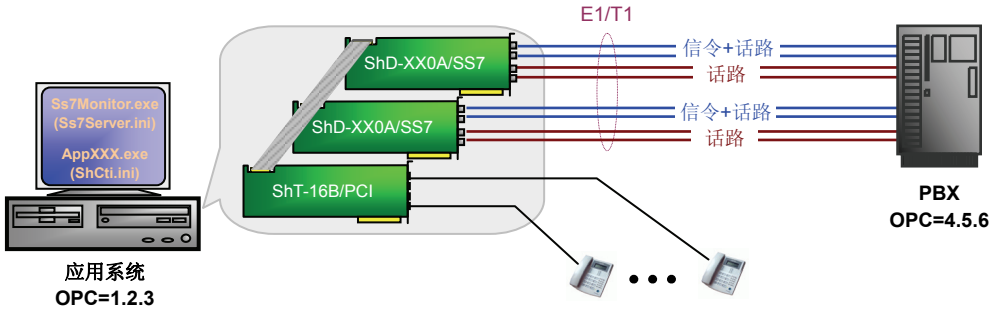


假设上述系统使用 SHD-60A-CT/SS7/PCI 板卡，第 1 条数字中继线的 16 时隙作为信令链路，其余 30 个时隙作为语音信道，第 2 条数字中继线的 30 个时隙均为话路，16 时隙不使用。以 TUP 协议为例，系统的配置为：

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=1.2.3 ServerIP=127.0.0.1 **[Ss7ClientInfo]** MaxSs7Client=1 IP[0]=127.0.0.1 **[Ss7PcmLinkInfo]** MaxSs7Pcm=1 Ss7PcmLink[0]=IP[0],LocalPCM[0]	…… **[SS7]** Ss7ServerIP=127.0.0.1　　//服务器的 IP 地址 LocalIP=127.0.0.1　　//本机的 IP 地址 **[PcmInfo]** TotalPcm=2 Pcm[0]=0,0 Pcm[1]=0,1 **[BoardId=0]** ……

[LinkSetInfo]
 	
MaxLinkSet=1
LinkSet[0]=Ss7PcmLink[0]
 	
[DPCInfo]
 	
MaxDPC=1
DPC[0]=4.5.6,LinkSet[0]
 	
[TUPRouter]
 	
DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1]
	PcmNumber=2
	
	PcmSSx[0]=7 //使用 SS7 信令方式
	PcmClockMode[0]=0 //主时钟，线路同步方式
	Ss7SignalingTS[0]=16 //16 时隙作为信令时隙
	UseTS16AsCircuit[0]=0
	
	PcmSSx[1]=7 //使用 SS7 信令方式
	PcmClockMode[1]=2 //从时钟
	Ss7SignalingTS[1]=16 //16 时隙作为信令时隙
	UseTS16AsCircuit[1]=1

4.4.1.2 单机多卡的应用系统

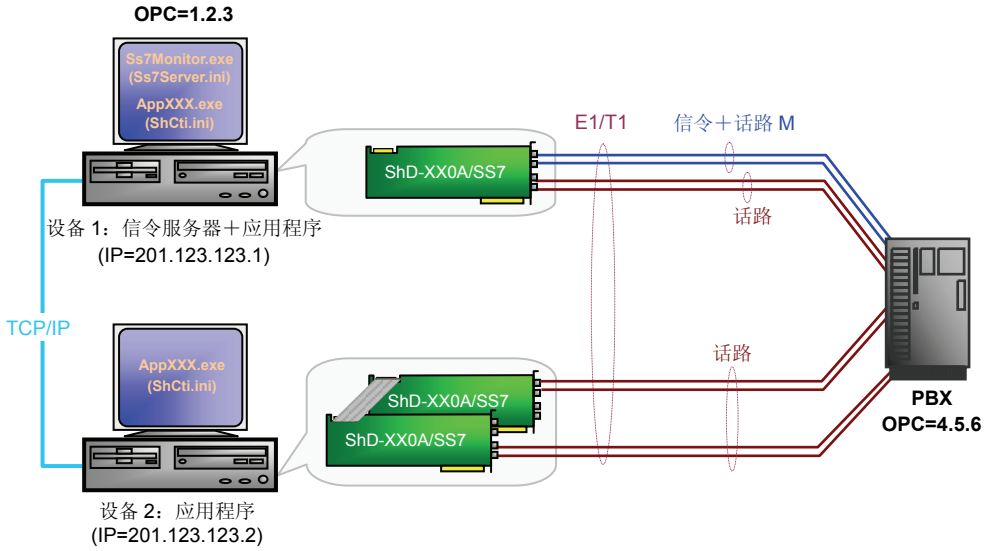


假设该系统包含 4 条 E1/T1 链路（由 2 块卡组成），1 块 16 路坐席卡，第 1 块卡的第 1 个 PCM 为信令/话路复用链路（信令时隙为 16 时隙），第 2 条 PCM 为纯话路；第 2 块卡第 1 个 PCM 为信令/话路复用链路（信令时隙为 16 时隙），第 2 条 PCM 为纯话路，以 TUP 为例，配置方法为：

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=1.2.3 ServerIP=127.0.0.1 **[Ss7ClientInfo]** MaxSs7Client=1 IP[0]=127.0.0.1 [**Ss7PcmLinkInfo**] MaxSs7Pcm=2 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[2] **[LinkSetInfo]** MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0]+Ss7PcmLink[1] **[DPCInfo]** MaxDPC=1 DPC[0]=4.5.6,LinkSet[0] **[TUPRouter]** DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1] DPC[0]:CIC_PCM[2]=IP[0],LocalPCM[2] DPC[0]:CIC_PCM[3]=IP[0],LocalPCM[3]	…… **[SS7]** Ss7ServerIP=127.0.0.1 LocalIP=127.0.0.1 **[PcmInfo]** TotalPcm=4 Pcm[0]=0,0 Pcm[1]=0,1 Pcm[2]=1,0 Pcm[3]=1,1 **[BoardId=0]** …… PcmNumber=2 PcmSSx[0]=7 PcmClockMode[0]=0 //主时钟，线路同步方式 PcmSSx[1]=7 PcmClockMode[1]=2 //从时钟 **[BoardId=1]** …… PcmNumber=2 PcmSSx[0]=7 PcmClockMode[0]=2 //从时钟 PcmSSx[1]=7 PcmClockMode[1]=2 //从时钟

	[BoardId=2] //设置 SHT-16B-CT/PCI ……

4.4.1.3 多机系统



假设应用系统系统包括 2 台设备，设备 1 上同时运行 SS7 信令服务器和应用程序，设备 2 为设备 1 的客户端。设备 1 包含 2 条数字中继线，其中，第 1 条数字中继线的 16 时隙为 SS7 信令链路，其余 30 个时隙为语音信道；第 2 条数字中继线的 16 时隙不使用，其余 30 个时隙为语音信道。设备 2 包含 2 条数字中继线，全部 30×2 个时隙都为语音信道。以 TUP 为例，配置方法如下：

➢ 设备 1：信令服务器和应用程序的配置文件

Ss7Server.ini	ShConfig.ini

[Ss7SystemConfig]
 	
OPC=1.2.3
ServerIP=201.123.123.1
 	
[Ss7ClientInfo]
 	
MaxSs7Client=2
IP[0]= 201.123.123.1
IP[1]= 201.123.123.2
 	
[Ss7PcmLinkInfo]
 	
MaxSs7Pcm=1
Ss7PcmLink[0]=IP[0],LocalPCM[0]
 	
[LinkSetInfo]
 	
MaxLinkSet=1
LinkSet[0]=Ss7PcmLink[0]
 	
[DPCInfo]
 	
MaxDPC=1
DPC[0]=4.5.6,LinkSet[0]
	……
	[SS7]
	
	Ss7ServerIP=201.123.123.1 //服务器的 IP 地址
	LocalIP=201.123.123.1 //本机的 IP 地址
	……

[TUPRouter] DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1] DPC[0]:CIC_PCM[2]=IP[1],LocalPCM[0] DPC[0]:CIC_PCM[3]=IP[1],LocalPCM[1]	

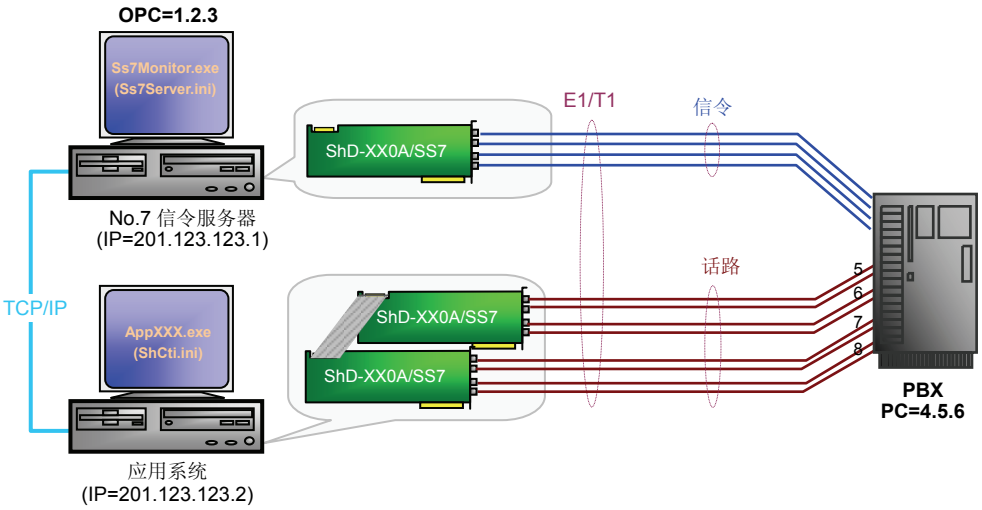
- 设备 2：应用程序的配置文件 ShConfig.ini

```
……
[SS7]

Ss7ServerIP=201.123.123.1          //服务器的 IP 地址
LocalIP=201.123.123.2              //本机的 IP 地址
……
```

4.4.1.4 使用专用信令服务器的多机系统

如果应用系统对可靠性要求比较高，建议单独使用一台计算机来处理信令，应用程序在其它的一台或多台计算机上运行，如下图所示：



上图中，设备 1 只处理信令，不处理话路，因此，信令服务器上除了运行 SS7 信令服务器，还需要为只提供信令链路的板卡加载 SynCTI 驱动程序；应用系统作为客户端。假设信令服务器包含 2 条数字中继线，只使用其中的 16 时隙；设备 2 包含 4 条数字中继线，电信局在 TUP 协议中为其分配的编号为 5、6、7、8。配置方法为：

- SS7 信令服务器的配置

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=1.2.3 ServerIP=201.123.123.1　　//信令服务器的 IP 地址 **[Ss7ClientInfo]** MaxSs7Client=2 IP[0]= 201.123.123.1 IP[1]= 201.123.123.2 **[Ss7PcmLinkInfo]** MaxSs7Pcm=2 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[1]	…… **[SS7]** Ss7ServerIP=201.123.123.1　　//服务器的 IP 地址 LocalIP=201.123.123.1　　//客户端程序的 IP 地址 ……

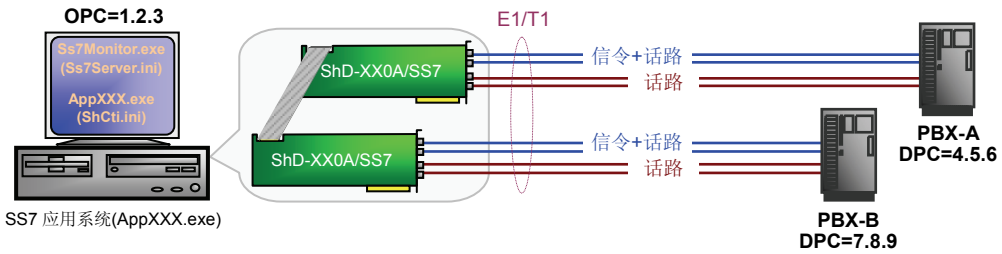
[LinkSetInfo] MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1] **[DPCInfo]** MaxDPC=1 DPC[0]=4.5.6,LinkSet[0] **[TUPRouter]** DPC[0]:CIC_PCM[5]=IP[1],LocalPCM[0] DPC[0]:CIC_PCM[6]=IP[1],LocalPCM[1] DPC[0]:CIC_PCM[7]=IP[1],LocalPCM[2] DPC[0]:CIC_PCM[8]=IP[1],LocalPCM[3] **[Monitor]** ConfigAsGateway=1 //需要加载 SynCTI 驱动程序	

➢ 应用系统的配置文件 ShConfig.ini

……
[SS7]

Ss7ServerIP=201.123.123.1 //服务器的 IP 地址
LocalIP=201.123.123.2 //本机的 IP 地址
……

4.4.2 单 OPC/多 DPC



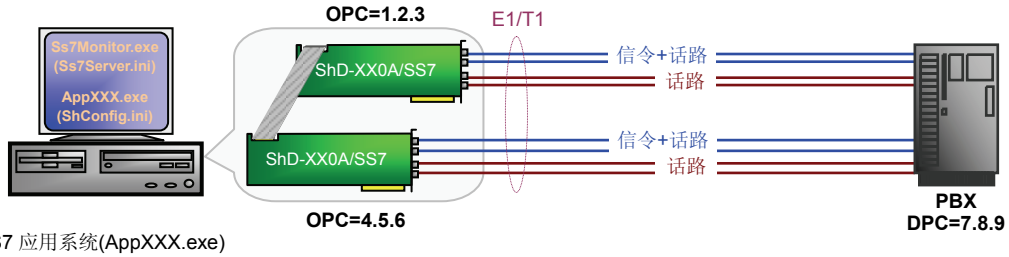
假设应用系统包含 2 片 SHD-60A-CT/PCI/SS7 板卡。每片板卡上的 2 条数字中继线的时隙 1~15、17~31 均为语音信道，第 1 条数字中继线上的第 16 时隙为信令链路，第 2 条数字中继线上的第 16 时隙不使用。2 片板卡分别连接到了具有不同 DPC 的 2 台交换机。以 TUP 为例，配置方法为：

Ss7Server.ini	ShConfig.ini

[Ss7SystemConfig]
 	
OPC=1.2.3
ServerIP=127.0.0.1
 	
[Ss7ClientInfo]
 	
MaxSs7Client=1
IP[0]= 127.0.0.1 //客户端程序的 IP 地址
 	
[Ss7PcmLinkInfo]
 	
MaxSs7Pcm=2
Ss7PcmLink[0]=IP[0],LocalPCM[0]
Ss7PcmLink[1]=IP[0],LocalPCM[2]
 	
[LinkSetInfo]
 	
MaxLinkSet=2
LinkSet[0]=Ss7PcmLink[0]
LinkSet[1]=Ss7PcmLink[1]
 	
[DPCInfo]
 	
MaxDPC=2
	……
	
	[PcmInfo]
	
	TotalPcm=4
	Pcm[0]=0,0
	Pcm[1]=0,1
	Pcm[2]=1,0
	Pcm[3]=1,1
	
	[BoardId=0]
	
	……
	PcmNumber=2
	
	PcmSSx[0]=7
	PcmSSx[1]=7
	PcmClockMode[0]=0 //主时钟，线路同步方式
	PcmClockMode[1]=2 //从时钟
	
	[BoardId=1]
	
	……

DPC[0]=4.5.6,LinkSet[0]
DPC[1]=7.8.9,LinkSet[1]
[TUPRouter]
DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1]
DPC[1]:CIC_PCM[0]=IP[0],LocalPCM[2]
DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[3]
	PcmNumber=2
	PcmSSx[0]=7
	PcmSSx[1]=7
	PcmClockMode[0]=2 //从时钟
	PcmClockMode[1]=2 //从时钟

4.4.3 多 OPC 的 SS7 应用系统



SS7 应用系统(AppXXX.exe)

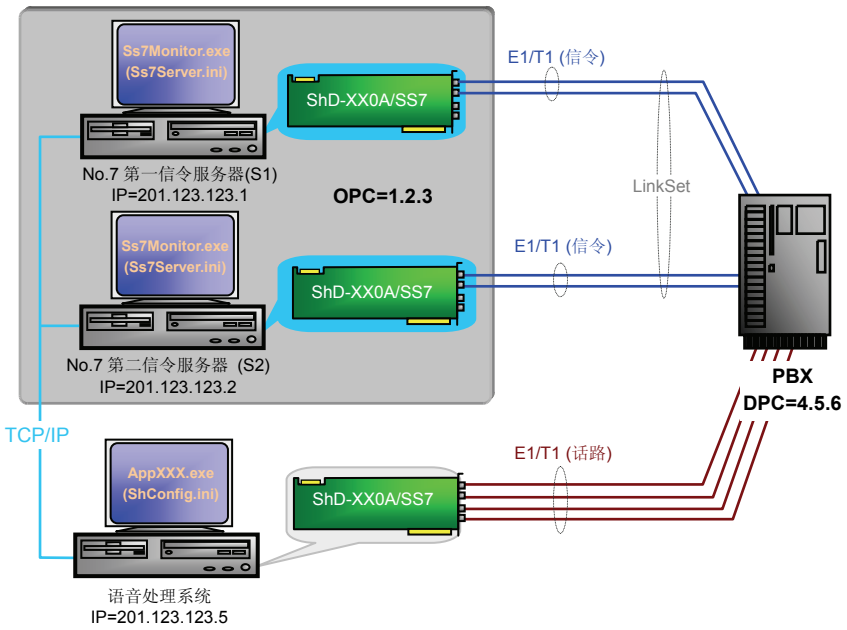
假设应用系统包含 2 片 SHD-60A-CT/PCI/SS7 板卡。第 1 个 PCM 为信令/话路复用链路(信令时隙为 16 时隙)，第 2 个 PCM 为纯话路，第 1、2PCM 对应一个 OPC，第 3 个 PCM 为信令/话路复用链路（信令时隙为 16 时隙），第 4 个 PCM 为纯话路，第 3、4PCM 对应一个 OPC。以 TUP 协议为例，系统的配置方法为：

Ss7Server.ini	ShConfig.ini

[Ss7SystemConfig]
OPC=1.2.3
ServerIP=127.0.0.1
[Ss7ClientInfo]
MaxSs7Client=1
IP[0]=127.0.0.1
[Ss7PcmLinkInfo]
MaxSs7Pcm=2
Ss7PcmLink[0]=IP[0],LocalPCM[0]
Ss7PcmLink[1]=IP[0],LocalPCM[2]
[LinkSetInfo]
MaxLinkSet=2
LinkSet[0]=Ss7PcmLink[0],1.2.3
LinkSet[1]=Ss7PcmLink[1],4.5.6
[DPCInfo]
MaxDPC=2
DPC[0]=7.8.9,LinkSet[0]
DPC[1]=7.8.9,LinkSet[1]
[TUPRouter]
DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1]
DPC[1]:CIC_PCM[0]=IP[0],LocalPCM[2]
DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[3]
	……
	[PcmInfo]
	TotalPcm=4
	Pcm[0]=0,0
	Pcm[1]=0,1
	Pcm[2]=1,0
	Pcm[3]=1,1
	[BoardId=0]
	……
	PcmNumber=2
	PcmSSx[0]=7
	PcmSSx[1]=7
	PcmClockMode[0]=0 //主时钟，线路同步方式
	PcmClockMode[1]=2 //从时钟
	[BoardId=1]
	……
	PcmNumber=2
	PcmSSx[0]=7
	PcmSSx[1]=7
	PcmClockMode[0]=2 //从时钟
	PcmClockMode[1]=2 //从时钟

4.4.4 高可靠性的应用系统

Synway 推出的 SS7 信令服务器支持双机热备份的运行方式，可以为应用系统提供电信级的可靠性。一个典型的高可靠性应用系统如下图所示：



上图中，SS7 信令服务器有 2 台，每台均安装一片 SHD 系列板卡，但只使用数字中继线的信令时隙，不处理话路。2 台服务器同时工作，共用一个点码（即 OPC=1.2.3），为语音处理系统提供信令服务，其中一台作为主服务器(Master)，另外一台作为从服务器(Slave)。当主服务器出现故障时，从服务器会自动接管全部的信令处理业务。

主、从信令服务器的确定规则如下：

1、 先启动的信令服务器为主服务器，后启动的信令服务器为从服务器。
2、 主、从服务器均正常工作时，如果主服务器出现故障，从服务器自动转变为主服务器，待原主服务器故障恢复后再次投入使用时，会自动变为从服务器。
3、 主、从服务器均正常工作时，可在主/从服务器之间进行手工切换，切换结束后，原主服务器变为从服务器，原从服务器变为主服务器。

语音处理系统由一台或多台安装 SHD 系列板卡的计算机组成，通过局域网从服务器获得信令服务，因此，客户端计算机上无需运行 SS7 信令服务器程序。

➢ 第一信令服务器(简称为 S1)的配置（包括 Ss7Server.ini 和 ShConfig.ini）

Ss7Server.ini	ShConfig.ini

[Ss7SystemConfig]
 	
OPC=1.2.3
ServerIP=201.123.123. 1 //本服务器的 IP 地址
SecondServerIP=201.123.123.2 //S2 的 IP 地址
……
 	
[Ss7ClientInfo]
 	
MaxSs7Client=3
IP[0]=201.123.123.1 //信令链路处理程序
IP[1]=201.123.123.2 //信令链路处理程序
IP[2]=201.123.123.3 //客户端程序的 IP 地址
 	
[Ss7PcmLinkInfo]
 	
MaxSs7Pcm=2
Ss7PcmLink[0]=IP[0],LocalPCM[0]
Ss7PcmLink[1]=IP[1],LocalPCM[0]
	……
	[PcmInfo]
	
	TotalPcm=1
	Pcm[0]=0,0
	[BoardId=0]
	
	……
	PcmNumber=1
	PcmSSx[0]=7
	PcmClockMode[0]=0 //主时钟，线路同步方式
	
	[SS7]
	
	Ss7ServerIP=201.123.123.1 //S1 的 IP 地址
	SecondServerIP=201.123.123.2 //S2 的 IP 地址
	LocalIP= 201.123.123.1 //本机的 IP 地址
	……

[LinkSetInfo]
 	
MaxLinkSet=1
LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1]
 	
[DPCInfo]
 	
MaxDPC=1
DPC[0]=4.5.6,LinkSet[0]
 	
[TUPRouter] //如不使用 TUP 协议则无需配置
 	
DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1]
 	
[ISUPRouter] //如不使用 ISUP 协议则无需配置
 	
DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1]
 	
[Monitor]
 	
ConfigAsGateway=1 //需要加载 SynCTI 驱动程序
……	

➢ 第二信令服务器(简称为 S2)的配置（包括 Ss7Server.ini 和 ShConfig.ini）

Ss7Server.ini	ShConfig.ini

[Ss7SystemConfig]
 	
OPC=1.2.3
ServerIP=201.123.123.2 //本服务器的 IP 地址
SecondServerIP=201.123.123.1 //S1 的 IP 地址
……
[Ss7ClientInfo]
MaxSs7Client=3
IP[0]= 201.123.123.1 //信令链路处理程序
IP[1]= 201.123.123.2 //信令链路处理程序
IP[2]= 201.123.123.3 //客户端程序的 IP 地址
 	
[Ss7PcmLinkInfo]
 	
MaxSs7Pcm=2
Ss7PcmLink[0]=IP[1],LocalPCM[0]
Ss7PcmLink[1]=IP[0],LocalPCM[0]
 	
[LinkSetInfo]
 	
MaxLinkSet=1
LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1]
 	
[DPCInfo]
 	
MaxDPC=1
DPC[0]=4.5.6,LinkSet[0]
 	
[TUPRouter] //如不使用 TUP 协议则无需配置
 	
DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1]
 	
[ISUPRouter] //如不使用 ISUP 协议则无需配置
 	
DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0]
DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1]
 	
[Monitor]
 	
ConfigAsGateway=1 //需要加载 SynCTI 驱动程序
……
	……
	[PcmInfo]
	
	TotalPcm=1
	Pcm[0]=0,0
	[BoardId=0]
	
	……
	PcmNumber=1
	PcmSSx[0]=7
	PcmClockMode[0]=0 //主时钟，线路同步方式
	
	[SS7]
	
	Ss7ServerIP=201.123.123.1 //S1 的 IP 地址
	SecondServerIP=201.123.123.2 //S2 的 IP 地址
	LocalIP= 201.123.123.2 //本机的 IP 地址
	……

➢ 客户端程序的配置（ShConfig.ini）

……

[PcmInfo]

TotalPcm=1
Pcm[0]=0,0

[BoardId=0]

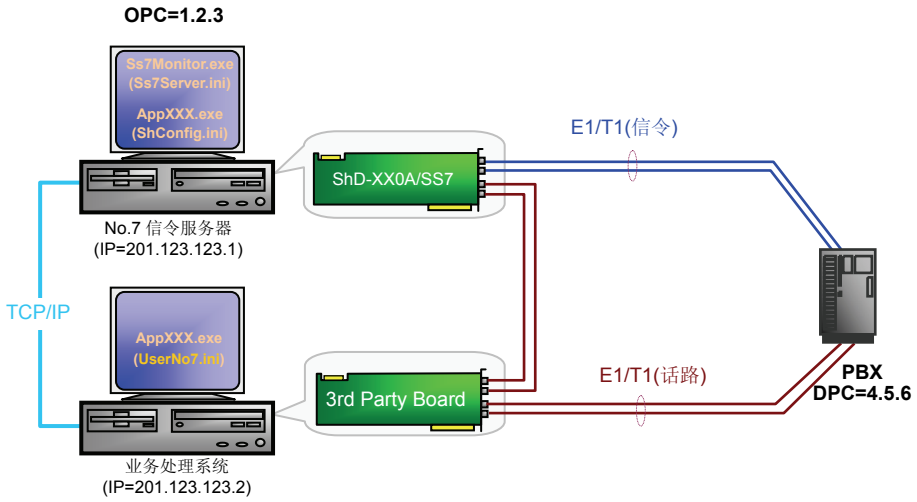
……

PcmNumber=2
PcmSSx[0]=7
PcmSSx[1]=7
PcmClockMode[0]=0 //主时钟，线路同步方式
PcmClockMode[1]=2 //从时钟

[SS7]

Ss7ServerIP=201.123.123.1 //S1 的 IP 地址
SecondServerIP=201.123.123.2 //S2 的 IP 地址
LocalIP=201.123.123.3 //本机的 IP 地址
……

4.4.5 为第三方板卡提供 SS7 信令服务



上图中，SS7 信令服务器使用Synway板卡处理信令，业务处理系统使用第三方板卡，信令信息通过TCP/IP协议从服务器获得。假设接入到SS7 信令服务器的数字中继线中还包含语音时隙，可以通过板卡的语音交换功能，通过设置配置项LoadShp_a3AsSIU，将语音时隙交换到另一个数字中继线接口，并连接到应用系统，如图中虚线所示。以TUP协议为例，系统的配置方法为：

- SS7 信令服务器的配置（Ss7Server.ini 和 ShConfig.ini）

Ss7Server.ini	ShConfig.ini

[Ss7SystemConfig]
OPC=1.2.3
ServerIP=201.123.123.1
[Ss7ClientInfo]
MaxSs7Client=2
IP[0]= 201.123.123.1
IP[1]= 201.123.123.2
[Ss7PcmLinkInfo]
MaxSs7Pcm=1
Ss7PcmLink[0]=IP[0],LocalPCM[0]
	……
	[SystemConfig]
	LoadShp_a3AsSIU=1 //输出语音时隙
	……
	[PcmInfo]
	TotalPcm=1
	Pcm[0]=0,0
	[BoardId=0]
	PcmNumber=1

[LinkSetInfo] MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0] **[DPCInfo]** MaxDPC=1 DPC[0]=4.5.6,LinkSet[0] **[TUPRouter]** DPC[0]:CIC_PCM[0]=IP[1],LocalPCM[0] **[Monitor]** ConfigAsGateway=1 //需要加载 SynCTI 驱动程序	PcmSSx[0]=7 PcmClockMode[0]=0 //主时钟，线路同步方式

➢ 基于第三方板卡的应用程序使用 SS7 信令服务器的客户端 SDK 时的配置文件 UserNo7.ini：

[SystemConfig]

TotalPcm=2
DefaultPcmLinkStatus=0x0000
……

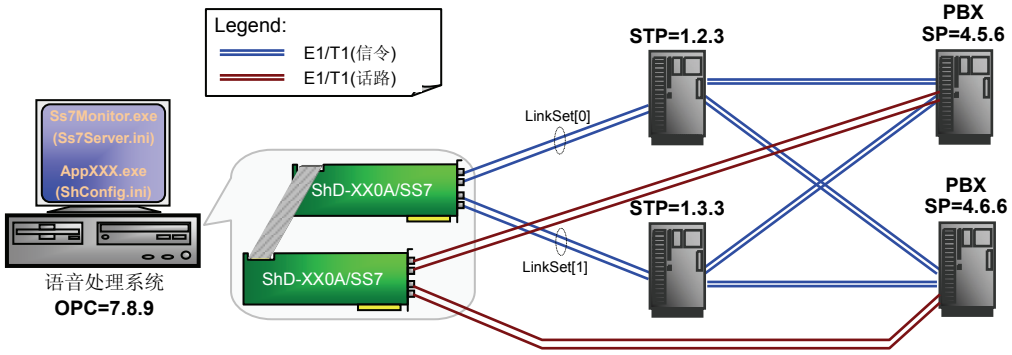
[AppChToPcmTsTable]

TotalAppCh=60
AppCh[0]=0,1..15
AppCh[15]=0,17..31
AppCh[30]=1,1..15
AppCh[45]=1,17..31

[SS7]

Ss7ServerIP=201.123.123.1 //信令服务器 IP
LocalIP=201.123.123.2 //本地 IP
……

4.4.6 准直连的应用系统



上图中，假设应用系统由 2 片 SHD-120A/SS7 板卡组成，第 1 片板卡只处理 4 条数字中继线上的信令（PCM 的逻辑编号为 0、1、2、3，并且信令时隙均为 16 时隙），第 2 片板卡处理 4 条数字中继线上的话路。以 TUP 为例，应用系统的配置如下：

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=7.8.9 ServerIP=127.0.0.1 **[Ss7ClientInfo]** MaxSs7Client=1 IP[0]=127.0.0.1	…… **[PcmInfo]** TotalPcm=8 Pcm[0]=0,0 Pcm[1]=0,1 Pcm[2]=0,2 Pcm[3]=0,3 Pcm[4]=1,0 Pcm[5]=1,1

[Ss7PcmLinkInfo]

MaxSs7Pcm=4
Ss7PcmLink[0]=IP[0],LocalPCM[0]
Ss7PcmLink[1]=IP[0],LocalPCM[1]
Ss7PcmLink[2]=IP[0],LocalPCM[2]
Ss7PcmLink[3]=IP[0],LocalPCM[3]

[LinkSetInfo]

MaxLinkSet=2

LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1]
LinkSet[1]=Ss7PcmLink[2]+ Ss7PcmLink[3]

[DPCInfo]

MaxDPC=2
DPC[0]=1.2.3,LinkSet[0]
DPC[1]=1.3.3,LinkSet[1]

[UP_DPCInfo]

MaxUP_DPC=2
UP_DPC[0]=4.5.6,LinkSet[0]+LinkSet[1] //负荷分担
UP_DPC[1]=4.6.6,LinkSet[1]+LinkSet[0],1 //第 1 组信令链路组为正常路由，第 0 路由为迂回路由

[TUPRouter]

UP_DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[4]
UP_DPC[0]:CIC_PCM[2]=IP[0],LocalPCM[5]
UP_DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[6]
UP_DPC[1]:CIC_PCM[2]=IP[0],LocalPCM[7]
……

Pcm[6]=1,2
Pcm[7]=1,3

[BoardId=0]

PcmNumber=4
PcmSSx[0]=7
PcmSSx[1]=7
PcmSSx[2]=7
PcmSSx[3]=7
PcmClockMode[0]=0 //主时钟，线路同步方式
PcmClockMode[1]=2 //从时钟
PcmClockMode[2]=2 //从时钟
PcmClockMode[3]=2 //从时钟
Ss7SignalingTS[0]=16 //16 时隙作为信令时隙
UseTS16AsCircuit[0]=0
Ss7SignalingTS[1]=16 //16 时隙作为信令时隙
UseTS16AsCircuit[1]=0
Ss7SignalingTS[2]=16 //16 时隙作为信令时隙
UseTS16AsCircuit[2]=0
Ss7SignalingTS[3]=16 //16 时隙作为信令时隙
UseTS16AsCircuit[3]=0
……

[BoardId=1]

PcmNumber=4
PcmSSx[0]=7
PcmSSx[1]=7
PcmSSx[2]=7
PcmSSx[3]=7
PcmClockMode[0]=2 //从时钟
PcmClockMode[1]=2 //从时钟
PcmClockMode[2]=2 //从时钟
PcmClockMode[3]=2 //从时钟
……

附录1 ISDN 通道释放原因值表

返回值	取值含义	支持协议
1	未分配的号码	DSS1, ISUP
2	无路由到达规定的转接网络	DSS1, ISUP
3	无路由到达终点	DSS1, ISUP
4	发送特殊的信息音	ISUP
5	转接前缀拨号错误	ISUP
6	不可接受的通路	DSS1, ISUP
7	呼叫已经给出并正在已建立的通路上递交	DSS1, ISUP
8	先占	DSS1, ISUP
9	先占电路留作重新使用	ISUP
16	正常的呼叫清除	DSS1, ISUP
17	用户忙	DSS1, ISUP
18	用户无响应	DSS1, ISUP
19	用户无应答（用户已提醒）	DSS1, ISUP
20	用户缺席	DSS1, ISUP
21	呼叫拒绝	DSS1, ISUP
22	号码变更	DSS1, ISUP
26	清除未选择的用户	DSS1, ISUP
27	终点故障	DSS1, ISUP
28	无效的号码格式	DSS1, ISUP
29	性能被拒绝	DSS1, ISUP
30	对 STATUS ENQUIRY 的响应	DSS1
31	正常未规定	DSS1, ISUP
34	无可用的电路/通路	DSS1, ISUP
38	网络失序	DSS1, ISUP
39	永久帧方式连接未开放业务	DSS1
40	永久帧方式连接可运营	DSS1
41	临时故障	DSS1, ISUP
42	交换设备拥塞	DSS1, ISUP
43	接入信息被丢弃	DSS1, ISUP
44	请求的电路/通路不可用	DSS1, ISUP
46	优先呼叫阻塞	DSS1, ISUP
47	资源不可用，未规定	DSS1, ISUP
49	业务质量不可用	DSS1
50	未预订所请求的性能	DSS1, ISUP
53	CUG 内呼出呼叫阻塞	DSS1, ISUP
55	CUG 内呼入呼叫阻塞	DSS1, ISUP
57	承载能力未认可	DSS1, ISUP

58	目前尚无可用的承载能力	DSS1, ISUP
62	分配的呼出接入信息与用户级别不一致	DSS1, ISUP
63	无适用的业务或任选项目，未规定	DSS1, ISUP
65	承载能力未实施	DSS1, ISUP
66	通路类型未实施	DSS1
69	请求的性能未实施	DSS1, ISUP
70	只有受限的数字信息承载能力可用	DSS1, ISUP
79	业务或任选未实施，未规定	DSS1, ISUP
81	无效的呼叫参考值	DSS1
82	所标识的通路不存在	DSS1
83	存在暂停的呼叫，但无所用的呼叫身份	DSS1
84	呼叫身份在使用	DSS1
85	无呼叫暂停	DSS1
86	具有所请求的呼叫身份的呼叫已被清除	DSS1
87	用户不是 CUG 成员	DSS1, ISUP
88	不兼容的终点	DSS1, ISUP
90	不存在的 CUG	DSS1, ISUP
91	无效的转接网络选择	DSS1, ISUP
95	无效的消息，未规定	DSS1, ISUP
96	必选信息单元丢失	DSS1
97	消息类型不存在或未实施	DSS1, ISUP
98	消息与呼叫状态不符或消息类型不存在 或未实施	DSS1
99	信息单元不存在或未实施	DSS1, ISUP
100	无效的信息单元内容	DSS1
101	消息与呼叫状态不符	DSS1
102	定时器超时的恢复	DSS1, ISUP
103	参数不存在或未实施，通过	ISUP
110	带不可识别参数的消息，丢弃	ISUP
111	协议差错，未规定	DSS1, ISUP
127	互通，未规定	DSS1, ISUP